# DEVELOPING AN ENVIRONMENT FOR EDUCATIONAL META-DATA:

# "MetaXML"

**A MASTER'S THESIS**

**in**

**Computer Engineering**

**Atılım University**

**by**

**BURCU ÖZDOĞRU**

**JUNE 2006**

# DEVELOPING AN ENVIRONMENT FOR EDUCATIONAL META-DATA:

# "MetaXML"

**A THESIS SUBMITTED TO**

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

**OF**

**ATILIM UNIVERSITY**

**BY**

**BURCU ÖZDOĞRU**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF**

**MASTER OF SCIENCE**

**THE DEPARTMENT OF COMPUTER ENGINEERING**

**JUNE 2006**

Approval of the Graduate School of Natural and Applied Sciences

________________________

Prof. Dr. Selçuk Soyupak

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science/Arts.

________________________

Prof. Dr. İbrahim Akman

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science/Arts.

________________________

Asst. Prof. Dr. Nergiz Ercil Çağıltay

Supervisor

Examining Committee Members

Prof. Dr. İbrahim Akman ________________________

Assoc. Prof. Dr. Kürşat Çağıltay ________________________

Asst. Prof. Dr. Çiğdem Turhan ________________________

Asst. Prof. Dr. Nergiz Ercil Çağıltay ________________________

Instructor Gül Tokdemir ________________________

# ABSTRACT

## DEVELOPING AN ENVIRONMENT FOR EDUCATIONAL META-DATA: "MetaXML"



Özdoğru, Burcu

M.S., Computer Engineering Department

Supervisor: Asst. Prof.Dr. Nergiz Ercil Çağıltay



June 2006, 71 pages



The purpose of this thesis is to provide a solution for some of the problems that occur related with educational meta-data. For defining educational materials and sharing it among different educational systems, educational meta-data standards should be employed. However, for the non-technical people defining the meta-data information of an educational material and then querying the related materials according to their educational needs are the main problems. This thesis reviews the educational meta-data standards and introduces a tool named metaXML. metaXML is a tool which is developed for educational meta-data operations by supporting SCORM object model. metaXML is based on a relational database system and provides an easy interface for the educators to define the meta-data of the educational materials. It also helps the educators to better address the educational materials by providing a query interface on the defined meta-data structure.



Keywords: Educational Meta-data, Learning Object Metadata, Relational Database, metaXML

# ÖZ


**ÖĞRENME SİSTEMLERİ ÜST VERİSİ TANIMLANMASI İÇİN YENİ BİR ORTAM GELİŞTİRİLMESİ : MetaXML**

Özdoğru, Burcu

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Nergiz Ercil Çağıltay

Haziran 2006, 71 sayfa

Bu tezin amacı, öğretimsel üst verilerde kaynaklanan bir takım problemlere çözüm üretmektir. Öğretimsel materyallerin değişik öğrenim sistemlerinde tanımlanmasında ve paylaşılmasında öğretimsel üst veri standartlarının önemli bir rolü vardır. Öğretimsel bir materyalde üst veri bilgisi tanımlamak ve ilgili materyallerde ihtiyaçları doğrultusunda sorgulama yapmak isteyen ancak yeterli teknik bilgisi olmayan eğitmenler için bu durum genel bir sorun oluşturmaktadır. Bu tez öğretimsel üst veri standartlarını yeniden incelemekle beraber metaXML adında bir sistemi anlatmaktadır. metaXML, SCORM obje modellemesinin yardımıyla öğretimsel üst veri işlemleri için geliştirilmiş bir programdır. metaXML, bağıntısal veritabanı baz alınarak geliştirilmiş, eğitmenlerin öğretimsel materyallerde üst verilerin tanımlanmasını kolaylıkla sağlayan bir arayüz programıdır. Aynı zamanda sistemde tanımlanan üst veri yapılarında sorgulama yapılmasını sağlayan bir arayüz de geliştirilmiştir.



Anahtar Kelimeler: Öğretimsel Üst Verisi, Öğrenme Sistemleri Üst Verisi, Bağlantısal Veritabanı, metaXML

To My Parents

## ACKNOWLEDGMENTS

I express sincere appreciation to my supervisor Asst. Prof.Dr. Nergiz Ercil Çağıltay for her guidance and insight throughout this research.

I would like to express my appreciation especially to my parents who are always there for me and my friends who always support my studies.

## TABLE OF CONTENTS

# LIST OF FIGURES

**FIGURES**

## LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| ADL | - | Advanced Distributed Learning |
| DCMI | - | Dublin Core Meta-data Initiative |
| DoD | - | Department of Defense |
| DTD | - | Document Type Definition |
| E-R | - | Entity Relationship Diagram |
| HTML | - | Hypertext Markup Language |
| IEEE | - | Institute of Electrical and Electronics Engineers |
| LMS | - | Learning Management System |
| LOM | - | Learning Object Meta-data |
| RDF | - | Resource Description Framework |
| SCO | - | Sharable Content Object |
| SCORM | - | Sharable Content Reference Model |
| SQL | - | Structured Query Language |
| URI | - | Uniform Resource Identifier |
| WWW | - | World Wide Web |
| W3C | - | World Wide Web Consortium |
| XML | - | Extensible Markup Language |
| XSD | - | XML Schema |

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction

Nowadays, with the rapid development of the Internet, distance education and e-learning programs are becoming more vital in educational world. E-learning alternatives make it possible for anybody to reach the educational resources and consequently to reach better educational goals. Actually, there are many ways of benefiting from distance education. Students, educators and people who are interested in e-learning can use the World Wide Web's resources, and if we think Internet as the world's biggest library, the resources are endless.

However, there are too many aspects to be examined in distance education programs. It is a new and fast growing industry, so for the time being there are many assumptions, thoughts and discussions in this area. First of all, we must have a clear thought of who may benefit from distance education. In fact, the answer is scary because anyone can use it. There are millions of educational institutions in the world and day by day, these institutions share a piece of the Internet by sharing their resources online. Distance education is not just about taking an educational degree; the starting point is to share any kind of information that is useful for students, teachers, thus in short the academic world. The aim is to reduce the workload on academics by sharing courseware and resources cross-institution. In other words, if academic A1 in the university U1 produces a courseware online and if there is no access restrictions then the academic A2 in the university U2 should use this existing resource. This leads to the term interoperability and actually it is really reasonable to use an existing resource rather than re-producing the material.

We have to understand that when there are so many educational institutions sharing their resources, the problems that we are facing grow in parallel. As an assumption, through the millions of colleges and universities, thousands of them teach, for example a course in introductory programming. Each programming course in each of these institutions describes, for example, arrays. Since it is a well defined topic, the description will be nearly the same with the other institutions. Eventually, we have thousands of similar description about arrays, and if these institutions offer online education then there are thousands of descriptions online. So, considering a student or teacher who searches for arrays on the net, s/he could not know which description is useful for him/her. Since the academic world is so enormous; consider the other consequences and the confusion it brings.

As a conclusion, the main problem of distance education programs is to share materials prepared for similar educational objectives. Depending on this problem, costs increase and efficiency decreases. Furthermore, because Internet is a huge library while searching for a subject, the user cannot be sure if the search results give the proper information or just garbage. Accordingly, we need to know the exact description of the resource. The Internet is full of worthless resources so it is nearly impossible to find the useful information we need because of the insufficiency of available searching technologies.

The solution is simple; try to use some data describing your content and try to do it in a way that other's can simply understand or adapt to their own educational content. This thought can be done by educational standards and specifications. Actually, the common property of all these standards and specifications is the meta-data [1][2]. Briefly, meta-data is data about data and it became important on the

World Wide Web because of the need to find useful information from the mess of information available. There is a relationship between meta-data and the information resource it describes. This means that, there is some information describing a resource which, we call it meta-data. As an example, a library catalog card record contains meta-data elements and these elements must be appropriate with certain standards. Also online systems for handling meta-data need to rely on their meta-data being predictable in both form and content. Predictability is assured only by conformance to standards.

There will always be a variety of meta-data standards. The advantage of using a meta-data standard is that your data will interoperate with others that use the same standard. Some standards have been developed to describe and provide access to a particular type of information resource. But within the education community, the two key parent standards are Dublin Core [3] and IMS [4].

IMS Global Consortium develops standards for educational systems. In the base of these standards we found Learning Object Meta-data (LOM) [1]. It is obvious that there is no problem with the LOM itself. Since LOM standard defines a set of meta-data elements that can be used to describe learning resources, LOM gives us what we need. The elements are categorized by the needs of the users and users define these meta-data for further usage of content packaging. By content packaging, researchers try to find solutions for educational systems [6]. With this solution, subjects can be used more efficiently. Content packaging finds a solution for a general problem but before making generalizations, there are some other problems based on LOM's architecture.

If we examine the architecture of LOM, we find Extensible Markup Language (XML) notifications. XML is a framework for defining markup languages. Since there is no fixed collection of markup tags, users may define their own tags according to their information. XML is designed to separate syntax from semantics to provide a common framework for structuring information with the features of platform independence and internalization support [7][8].

## 1.2 Statement of the Problem

While dealing with distance education systems, there are some steps to be followed by the implementers. The base part of these phases is to implement the meta-data which should be in conformity with the meta-data standards. After this phase, with content packaging the whole system can be developed. However, implementing the meta-data part is a complex process. Elements are implemented by XML codes. According to the characteristics of XML, it makes sense to build the LOM elements by using this structure. Another reason for implementing LOM in XML is the property of platform independence. Also, because of using plain text, it is easy to implement and use it in other platforms.

However, implementing this structure into software systems is a problem. Some developers prefer to use XML format directly. Unfortunately, using XML for meta-data descriptions limits the user's queries. It is not possible to create complex queries and build complex relations among the XML documents. Another problem is that, while defining the meta-data, the validation of the information should be checked and non-valid information should be extracted from the system. Because knowing XML is not a must for users, entering and retrieving data can cause some problems for the ordinary users. Additionally, to use XML, an editor must be used, so users must know how to handle an XML editor. These editors are general purpose editors (not specifically developed for educational purposes), for example they do not provide a list of available educational meta-data specifications [34]. Also, users cannot find sufficient programs for processing LOM elements for their actual needs. Educational meta-data tools are available with the ability of supporting one or more educational meta-data specifications. However, the tools which mainly focused on XML technologies have the disadvantage of not meeting with the educational needs such as not providing a list of available educational meta-data specifications. Also they require knowledge about XML [34]. Another important problem is the incapability of executing the search algorithms. The main reason for using search is to observe what the other users have done in the related subjects, choose the proximate result and adapt this result to their needs as well as discover the most appropriate learning object for their purposes. Especially in educational systems search is an important feature. By doing search in learning object meta-data, users can have many ideas for using their data. As an example, consider the web sites of elementary schools. In this web site, authorities want to put a picture of Atatürk. For this aim, they can search other web sites and figure out the size, format or description of other Atatürk pictures. These properties are meta-data characteristics. By using learning object meta-data, users can handle their web sites and find better answers for their questions. However, in order to create complex queries on top of the XML structure, users have to know the XML structure very well, and create their queries accordingly. This is another important problem for the ordinary users. Additionally, when the number of XML documents increase, the management of those documents becomes an important problem. In such cases organizations need to use a better system.

In recent years, for dealing with these problems of XML, important studies are in progress such as XML Databases [9][10] and XQuery [7][11]. It is obvious that through the extensive usage of XML and Internet, the way of exchanging and combining information has been affected thus, in order to catch the technology, various database vendors has added XML support. Since many applications that combine and exchange information also need to store it, XML is changing the way that databases are used.

XML has some advantages; if it is described in database format such as portability because of using Unicode, describing data in the structure of tree or graph and describing the structure and type names of the data. However, since accessing to the data is slow due to parsing and text conversion and having some incapability such as indexes, security, efficient storage, multi-user access, transactions and data integrity rules which can be used efficiently in relational databases shows us that XML databases are not mature enough to be used in some systems [9]. Main assumption of the XML databases is that, if data are stored as XML in databases, retrieving queries are much faster than the SQL queries defined for a relational database. While working with XML documents and a database for transferring data between them, mapping the XML document schema to the database schema is necessary. After mapping, in order to retrieve queries, XQuery technology can be used. XQuery is the language for querying XML data and a query in XQuery is an expression that reads a sequence of XML fragments or atomic values and returns a sequence of XML fragments or atomic values [11]. There are some requirements for XQuery. It must be declarative, protocol independent with respect to XML data modeling. Also, it must be able to transform and to create XML structures. Consequently, XQuery is designed for querying XML, but it is still in the draft phase and also for using this technology, we need XML tools. Nevertheless, XML databases and XQuery are bound to each other, so while storing XML data in an XML database we need XQuery for retrieving queries from the XML documents. However, the lack of querying across multiple documents in XML databases, does not give us a proper solution.

The required system should connect to XML and database and while this connection occurs, enable users to search over the database with specific conditions. Since the aim is to improve the sharing among users by using search; by using relational data modeling and a typical database management system, meta-data elements are stored in the database, and while doing search, users do not have to deal with XML, so the problems based on the structure of meta-data elements solved in a more efficient way.

This study proposes a platform (a meta-data editor) for meta-data sharing. The biggest problem of sharing meta-data is that, all elements are implemented by XML codes but this usage cannot be limited by the users who know XML. The goal of this study is to create a web based platform for educational meta-data based on IEEE LOM standard [5]. The platform creates the meta-data in a relational database and transforms XML documents into this model. It also creates the XML documents by querying information from the database. With this platform users can enter their data by using forms, and can see the results by means of forms or XML codes. The program is also supported with search algorithms in order to ensure the sharing among different users. The data is kept in the database to prevent usability and interoperability. The purpose is to provide sharable meta-data for people who use the same standards in their projects. The system also provides a platform for validating the information entered to the meta-data.

This study is organized as follows: Chapter 1 briefly introduces the topic and statement of the problem. Chapter 2 gives information about the background of the subject. Chapter 3 discusses the methodology used in continuation with the designed system. Finally, in Chapter 4 future studies and outcome of the thesis are mentioned.

# CHAPTER 2

# BACKGROUND INFORMATION AND LITERATURE REVIEW

In this Chapter, the meaning, usage and purpose of meta-data, educational meta-data standards, relationships between meta-data systems, the necessity of using learning object meta-data, the structure of meta-data and recent technological improvements for handling XML structure are mentioned.

## 2.1 Background

Meta-data is structured in a manner that facilitates the management, discovery and retrieval of resources on the World Wide Web (WWW). Meta-data standards have been developed to support both machine interoperability and targeted resource discovery by human users on the web. Meta-data standards for the Internet are an attempt to bridge the gap between the comprehensive cataloguing which is done by professionals in the library context. In particular, these meta-data standards allow creators of documents and managers of resource collections to describe resources in detailed manner facilitating targeted queries by search engines. A meta-data record typically consists of a set of elements or fields which describe in detail the content of the resource, its property rights and its instantiation.

The Dublin Core [3] is a set of standards for the use of the Resource Description Framework (RDF) dialect to XML, to describe library meta-data or any web page. Resource Description Framework is the specification for a meta-data model, often implemented as an application of XML, maintained by the W3C [12].

The purpose of Dublin Core Meta-data Initiative (DCMI) is to adopt the interoperable meta-data standards and to develop specialized meta-data vocabularies. According to the need to improve retrieval of information resources, especially on the World Wide Web, during 1995 and 1996, the Dublin Core meta-data element set was developed. This standard is mainly used by libraries, archives, government and other publishers of online information in order to provide the electronic resources they need [3].

Educational meta-data standards, such as the EdNa Meta-data standard [13], extend the scope of description that can be included in a meta-data record with information that has particular educational relevance. This is done by either defining education specific elements, element refinements or encoding schemas.

IMS [4] is a global consortium with members from educational, commercial and government organizations. It is developing and promoting open specifications for facilitating online distributed learning activities such as locating and using educational content, tracking learner progress, reporting learner performance and exchanging student records between administrative systems. It is concerned with standards for learning servers, learning content and the enterprise integration. The IMS initiative originated in higher education but stakeholders in corporate and government training, K-12 and continuing education have played an increasingly active role.

IMS has two key goals. The first one is defining the technical specifications for interoperability of applications and services in distributed learning. The second one is supporting the incorporation of the IMS specifications into products and services worldwide [14].

IMS is not building a software product. It is defining technical specifications that developers and creators of product and services can work together. It endeavors to promote the widespread adoption of specifications that will allow distributed learning environments and content from multiple authors [4].

It attempts to address the problems of the perceived lack of open standards in courseware or learning materials and the system that is needed to deliver these [15]. The aim is to provide standards for describing learning resources, communication protocols between learning resources, systems to manage the overarching delivery and handling of learning resources. This is known as learning management system (LMS) and researchers and developers are having an agreement on handling the learning objects (LO), which are the piece of information, by a learning management system [37]. LMS is a software application or web based technology used to plan, implement and assess a specific learning process and allows learners to authenticate themselves, register for courses, complete courses or take assessments. Typically a LMS provides an instructor with a way to create and deliver content, monitor student participation and assess student performance. It may also provide students with the ability to use interactive features such as threaded discussions, chat, mentoring, certification training, video conferencing and discussion forums. The Advanced Distance Learning group has created a set of specifications called Sharable Content Reference Model (SCORM) to encourage the standardization of LMS [16].

The IMS meta-data specification is derived from extensive collaborations, requirements meetings, focus groups and research related to the development of meta-data specification to support online learning. Groups included in the requirements process included teachers, instructional designers, digital library experts, administrators of educational institutions, software developers, content developers and meta-data experts. IMS meta-data will be represented in XML/RDF format.

To understand meta-data first we have to understand the meaning of data. On its own, data has no meaning, only when interpreted by some kind of data processing system it takes a meaning and become information.

Meta-data is data about data. As an example, library catalog card, which contains data about the nature and location of a book, can be considered as meta-data. The content combined with its meta-data is often called a content package, which is an encrypted file containing content and meta-data.

## 2.2 IMS Meta-data

When we follow a link or do-reference a URI, we reach something and usually we call this thing a resource. Sometimes it is referred to as a document because many of the things currently on the web are human readable documents [17]. One of the characteristics of the World Wide Web is that when you retrieve the resources, you reach information about that resource. This information is generally known as meta-data.

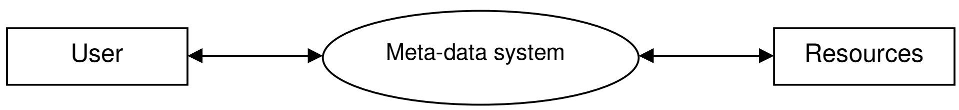


**Figure 2.1 Meta-data System [4]**

In Figure 2.1, we can see that meta-data system is between users and resources. Users can be learners, teachers etc. We can think this system as a connector from the both sides. Users discover resources and resources deliver to the users by the help of the system.  Without this system there will be no interoperability and resource discovery. For building such a system executives, catalogers and implementers are working together, as can be seen in Figure 2.2.

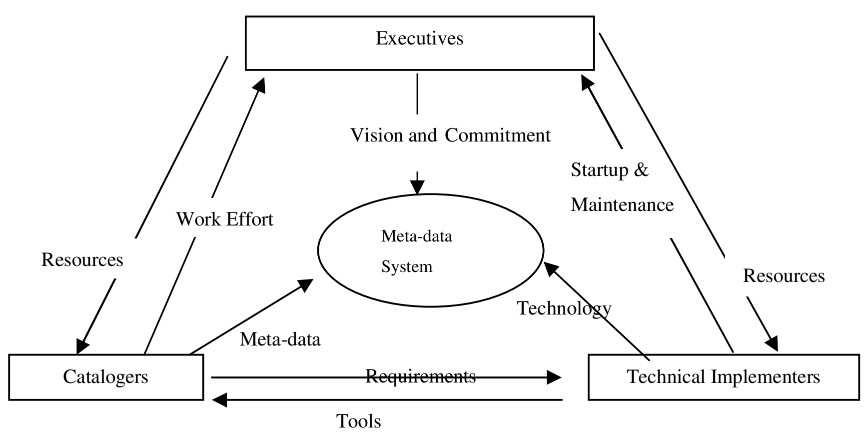


**Figure 2.2 – Relations between Users and Meta-data System [4]**

The meta-data system is about interoperability. It has the ability to exchange and interpret records, search other's meta-data and provide other's to search your meta-data. It requires adequate descriptions of resources, use well known definitions, structures and standard representation.

IMS used content packaging specification combining with IMS learning design as can be seen in Figure 2.3. The IMS Content Packaging Specification describes data structures that are used to provide interoperability of Internet based content with authoring tools, learning management systems and run-time environments. The objective is to define a standardized set of structures that can be used to exchange content. They are focused on defining interoperability between systems that wish to import, export, aggregate and disaggregate content packages. IMS content packages describe their contents in an XML document called the package manifest. The manifest include structured 'views' into the resources contained in that package and each 'view' is described as a hierarchy of items called 'organization'. Each item refers to a resource that, in turn, can refer to a physical file within the package [6].

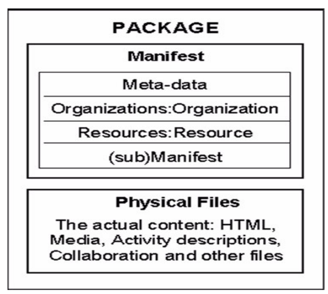


**Figure 2.3 – Structure of IMS Content Package [6]**

The IMS Content Packaging Conceptual Model defines a set of key elements that constitute the package. The IMS Manifest file describes the content organization and resources in the package and the actual physical files that are being packaged. The Manifest file is structured as Top-Level Manifest and for each instance of manifest contains the Meta-data section, Organization Section, Resources section and (sub) Manifest section [6].

The purpose of the Top-Level Manifest is to describe the package itself by using XML element. It may also contain optional sub-manifests. In meta-data section, an XML element describes a manifest as a whole. In organizations section, an XML element describes one or more organizations of the content within a manifest. In resources section, an XML element contains references to all of the actual resources and media elements needed for a manifest, including meta-data describing the resources and references to any external files. The (sub) manifests are logically nested manifests [6] [18].

In the content packaging specification, two types of meta-data are defined. There is a top-level meta-data section and there are the optional meta-data elements of the resources section. According to the IMS Content Packaging Best Practice Guide both sections are optional [19]. The first type of meta-data describes the manifest that contains it. The commonly used meta-data elements are title, description, keywords, a contributor's role, content's purpose (e.g., educational objective, skill level), and copyright. These meta-data elements should be drawn from the IMS Meta-data specification. If a meta-data element does not exist in the specification, it could be included by using an XML namespace in a manifest's meta-data element(s). All meta-data elements must be defined in a DTD, XDR or XSD that is declared at the top of the IMS Manifest File.

The Resources Section also can contain optional meta-data can be seen in Figure 2.4.

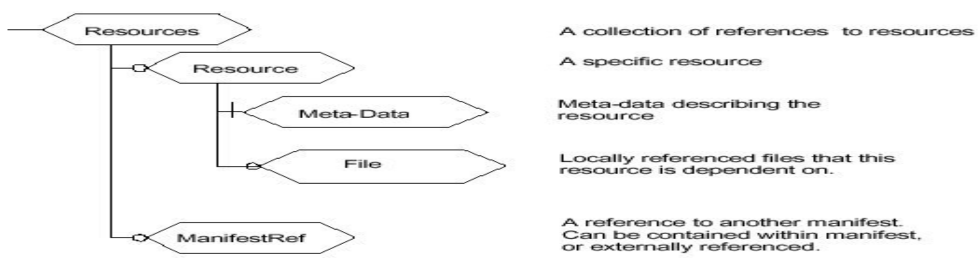


**Figure 2.4 – Resources Section of Manifest File [6]**

Content Packaging is another work to build a system. What we are dealing wiwith the meta-data part. As seen in the figures above, meta-data is a part of content packaging and by using the XML bindings of meta-data implementation; it becomes possible to build a system by using content packaging.

Manually created meta-data adds value because it ensures consistency. If one webpage about a topic contains a word or a phrase, then all web pages about that topic should contain that same word. It also ensures variety, so that if one topic has two names, each of these names will be used.

Used in the context of digital spatial data, meta-data is the background information which describes the content, quality, condition and other appropriate characteristics of the data. It serves many important purposes including data browsing, data transfer and data documentation [20].

Meta-data can be organized into several levels ranging from a simple listing of basic information about available data to detailed documentation about an individual data set. At a fundamental level, meta-data may support the creation of an inventory of the data holdings of a state or local government agency. At a more detailed level, it may be considered as insurance. It insures that potential data users can make an informed decision about whether data is appropriate for the intended use [21].

Actually the definition of meta-data differs from person to person but the common sense is that meta-data is data about data. If we look at this term from a programmer's point of view we reach some questions like 'what are the data fields in the database?', 'what are permissible values?', 'how are the values expressed?'. With these questions we reach another definition. Meta-data describes a data set and the data formats of the values.

An example of a data value can be a calendar date. There are several different standards and specifications for calendar dates. In order to interpret a block of data a programmer must know which one is being used [20].

As another example, think of a two dimensional matrix about high school students. The columns represent parameters and the rows represent the column values. Imagine in the first column we hold the weight of the students. The important thing we have to consider for the beginning is the data value of the weight such as integers or decimal values. How is a missing value encoded and so forth? All of this information about the data in the cell is the meta-data. It describes something about the data. After describing the data, we have to look at the entire table. The designers should determine the number of rows and columns and the titles of the columns. Actually this is the structure of the dataset and it is also meta-data. The entire dataset can also be considered as a resource. The data has an author or a publisher and this information is needed when searching for information. It is sometimes referred to as cataloging information or indexing. This information may have predefined descriptions of which information is to be included and how the information is to be represented. This is meta-data about the meta-data called as meta-meta-data.

To reach information we need resource. But how can we understand which resource is right for us. As we have mentioned before Internet is a huge library with relevant and irrelevant information. For example, consider you are searching for XML on the net, and you get 100 resources in return. It is really difficult to open all of these web pages in order to find what you are really looking for. We can think of resources as the containers of information. Meta-data provides information about the contents of the container without having to open the container. We can think meta-data as the label of the resource captures the basic characteristics of a data, which is usually presented as an XML document. It represents who, what, when, where, why and how of the resource. If a container has no label, then we cannot know what it is about. As a result, what is on the label reflects what is in the container. IMS provides a good set of labels which can be understood by meaning and representation. It also supports participation in the networked marketplace [20] [22].

IMS meta-data contains a core meta-data, standard extensions and custom extensions. In the labels, we see these elements, shown in Figure 2.5.

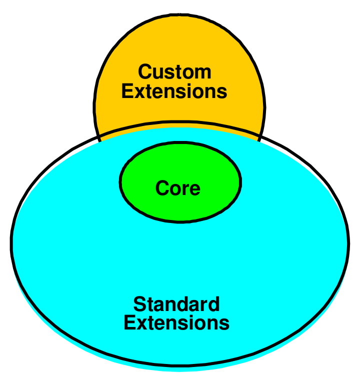


**Figure 2.5 – IMS Meta-data [4]**

### 2.2.1 IMS Core Meta-data

The IMS core meta-data contains [4]:

- Title
- Catalog
    - Entry
    - Source
- Language of the resource
- Description
- Version
- Contributor
    - Role
    - Entity
    - Date
- Meta-data scheme
- Meta-data language
- Format
- Location
- Cost
- Copyright and other
- Rights text
- Classification
    - Purpose
    - Description
    - Keywords

### 2.2.2 IMS Standard Extensions

A significant standard extension is the TaxonPath in the Classification structure. This element aggregates to trace the path set out in a structured taxonomy for any given term [1] [4].

- Classification
    - Purpose
    - Description
    - Keywords
    - *TaxonPath*

### 2.2.3 IMS Custom Extensions

A custom extension is a meta-data field that carries information that can not be carried in an existing meta-data field. IMS provides a well defined mechanism for creating extensions [1] [4].

### 2.2.4 The Structure of IMS Meta-data

Just like seen in the IEEE LOM conceptual model, the structure of the meta-data consists of categories and elements [5] [20].

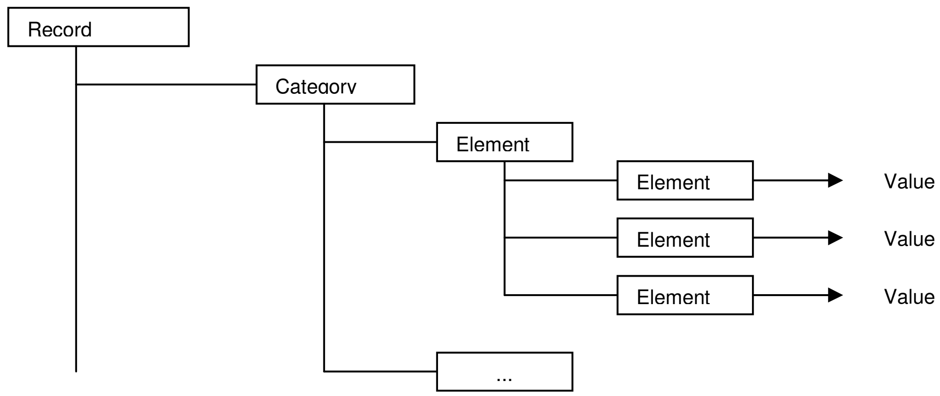


**Figure 2.6 – The Structure of IMS Meta-data [20]**

Figure 2.6 shows the basic structure of the IMS meta-data. The next thing is to fill the category, element and value parts with the appropriate values, as can be seen as an example in Figure 2.7.

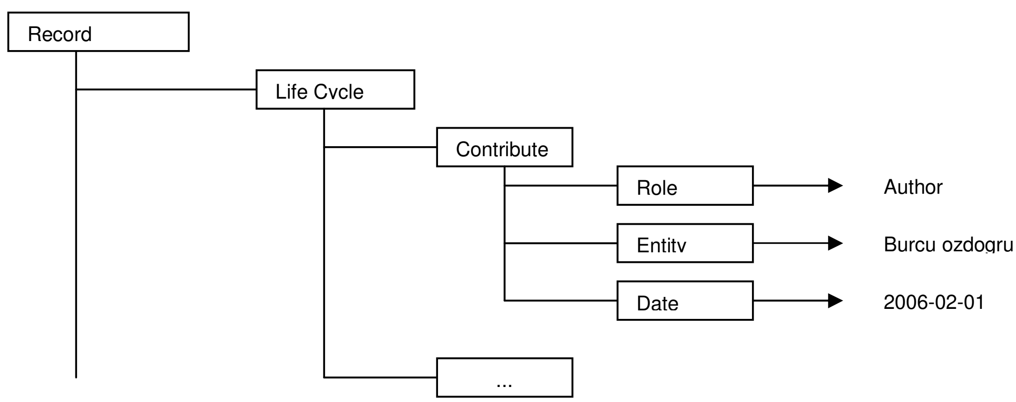


**Figure 2.7 – Example of IMS Meta-data Structure**

The elements must have clear definitions.

- Life Cycle. Characteristics related to the different phases
- Contribute: Involvement in the creation or edition
- Role: Kind of involvement
- Entity: Person or enterprise
- Date: A calendar date

This structure can be read as "This person (entity), as an author (role), contributes during the life cycle of the resource. All of the values must have data types. These data types are defined in the specification. Min-Max lengths (minimum length that must be supported) for data as strings are defined in the specification. Any field with a LangString data type may have multiple values, as can be seen in Figure 2.8.

```
LangString
        Language: tr
        String: kırmızı şapka
LangString
        Language: en
        String: red hat
```

**Figure 2.8 – LangString Data Type [4]**

### 2.2.5 Structured Meta-data in XML

Figure 2.9 shows the XML representation based on the given structure of meta-data.

```
Structure                      An XML Representation

General                        <GENERAL>
   Description                   <DESCRIPTION>
      LangString                    <LANGSTRING>
            String                      Operand conditioning simulation
                                    </LANGSTRING>
                                 </DESCRIPTION>
                               </GENERAL>
```

**Figure 2.9 – Structured Meta-data in XML [4]**

## 2.3 Relationships with other Meta-data Systems

IMS and Dublin Core are the two main organizations providing standards in educational systems however; they adapt other systems by means of providing solutions to the problems in learning object meta-data. In this section, the relations between these systems are mentioned.

### 2.3.1 IEEE Learning Technology Standards Committee Learning Object Meta-data (LOM)

Actually, IMS adopts the IEEE LTSC LOM [5]. The IEEE LOM standard defines a set of meta-data elements that can be used to describe learning resources. This includes the element names, definitions; data type's and field lengths. The standard is known as a multi-part standard and defines both a conceptual model for the meta-data and an XML binding. The standard includes conformance statements for how meta-data documents must be organized and how applications must behave in order to be considered IEEE conformant [5].

The IEEE conceptual data schema for meta-data definitions is hierarchical. At the base of the hierarchy is the "root" element containing many sub-elements. If a sub-element itself contains additional sub-elements it is called a "branch". Sub-elements that do not contain any sub-elements are called "leaves". This entire hierarchical model is called the "tree structure" of a document, as can be seen in Figure 2.10.

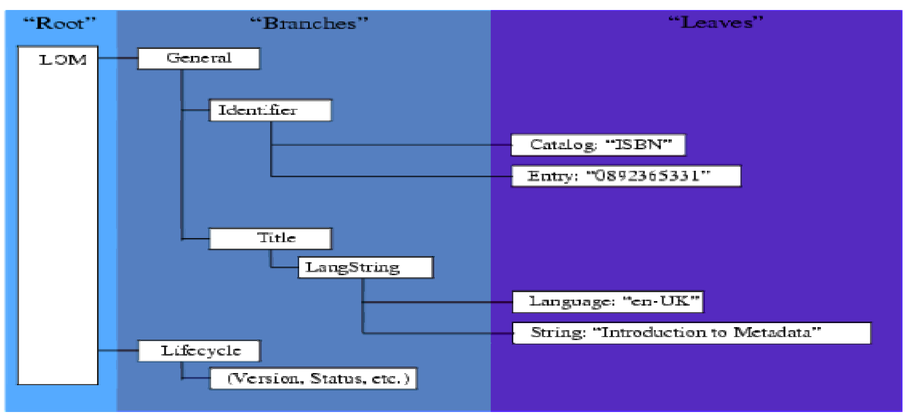


**Figure 2.10 – Root to Leaf "tree view" of Meta-data [1]**

Each element in the meta-data hierarchy has a specific definition, data type and value space. The IEEE LOM standard conceptual data schema lists all the meta-data elements in tabular format, shown in Figure 2.11.

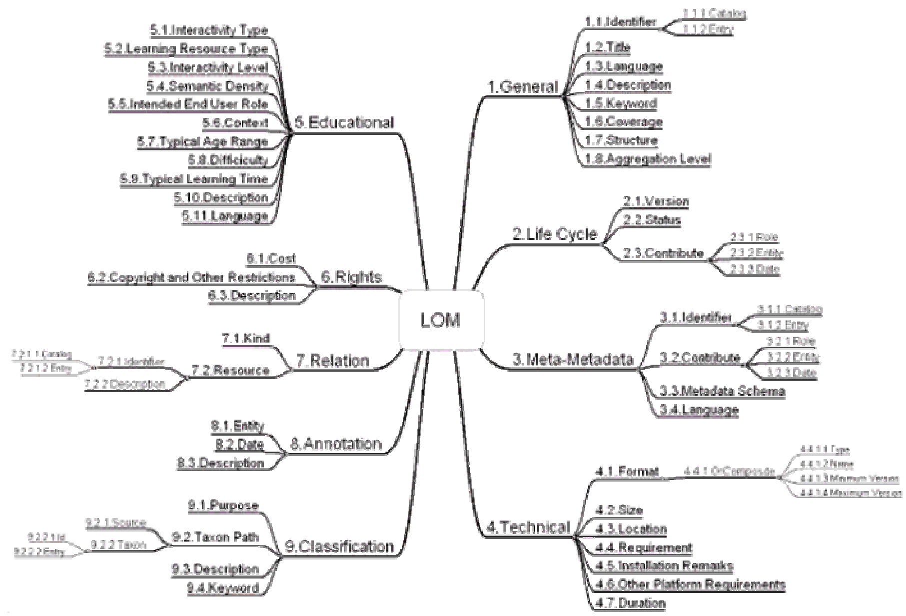


**Figure 2.11 – The Elements and Structure of the LOM Conceptual Data Schema [1]**

As shown in the Figure 2.11, the elements are divided into top nine categories named as General, Life Cycle, Meta-Meta-data, Technical, Educational, Rights, Relation, Annotation and Classification respectively. Each of these branches comprises several elements, some of which are leaves; others are sub-branches which lead to leaves.

#### 2.3.1.1 Advanced Distributed Learning (ADL) – Sharable Content Object Reference Model (SCORM)

Advanced Distributed Learning (ADL) is developed for military purposes for Department of Defense (DoD), to provide access to the highest quality education, training and performance aiding anytime and anywhere based on the needs of the personnel especially soldiers.

As taking the soldiers as a base of the whole system, ADL's aim is to increase the quality of education and reduce the cost that arises from the trainings in different locations. Soldiers can go anywhere for duty and their trainings are approximately similar, so in order to give the same training there must be a common way. Consider, for the soldiers who are in city A and B takes the same training but there are two different materials. It is a reasonable fact, if we do this for two or three cities but imagine this concept for the soldiers in 100 cities, then for the same training there are 100 different but almost the same material, so the purpose is to prepare only one material which offers all the necessary titles with a wide content.

Beginning from 1997, according to the needs of an interoperable, reusable and accessible system, ADL sets a strategy for DoD personnel. This strategy is to provide common, open, international specifications and standards. After some years of research, ADL introduces Sharable Content Object Reference Model (SCORM) [16].

The ADL initiative and SCORM seek to maximize technology-based learning to generate substantial cost savings. SCORM is designed for technical specification, for meeting technical requirements but is still an essential part of the ADL initiative.

SCORM can be used not only for DoD but also for all educational problems organizations or industry has, including distance education and e-learning. Before SCORM, there were some certain and constant problems of educational systems such as; not moving courses from one learning management system to another, not reusing content pieces across different courses, not sequencing reusable content for branching, remediation and other tailored learning strategies, not searching learning content libraries or media repositories across different learning management system environments. It is a fact that problems are similar; the important thing is to give education based on some standards according to the high level requirements [16].

In general, we can say that, the purpose of all such systems is to find the common areas used by the users, combine them and make them suitable with the learning management systems, and while doing this reduce the cost and increase the quality of education.

Every distributed learning environment needs essential high level attributes and SCORM ensures these attributes such as interoperability, accessibility, reusability, durability, maintainability and adaptability [16].

- **Interoperability:** the ability to take instructional components developed in one system and uses them in another system. (i.e. assume you develop content where the delivery platform is on a non-networked Macintosh, the aim is to ensure that this content will also operate on the web on a PC using both Internet Explorer and Netscape equally well.)
- **Durability:** the ability to withstand technology changes over time without costly redesign, reconfiguration or recoding. (i.e. upgrading an operating system from Windows 2000 to Windows XP must not have an impact on the delivery of content to the learner.)
- **Accessibility:** the ability to locate and access instructional components from multiple locations and deliver them to other locations. (i.e. a manager can conduct an online search for training on how to behave at work and identify appropriate materials for his/her specific organization needs based on information provided in the content meta-data.)
- **Reusability:** the ability to use instructional components in multiple applications, courses and contexts. (i.e. assume you have to prepare a course about Java Applets; with reusability you can use other people's sources without changing it.)
- **Maintainability:** the ability to withstand content evolution and changes without costly redesign, reconfiguration and recoding.
- **Adaptability:** the ability to change to satisfy differing user needs.

While working with the same system, there is no problem about using these attributes when combining the contents. Although the content can be anything, in many cases it must be specifically designed to fit with a particular LMS in order to work properly. However, content stops working when you try to migrate to other systems. SCORM achieved these attributes with the use of Sharable Content Objects (SCOs) composed of assets that launch in a SCORM runtime environment [23], [24].

Since there are many different learning management systems, each with its own strengths and weaknesses, content will not work properly with another learning management system, because that content was written specifically for a proprietary learning management system.

SCORM tries to give a solution to this problem. Every SCORM conformant LMS has a rigorously defined set of capabilities and behaviors; and with SCORM different LMS's interface with content in a carefully defined way [24].

The key concepts of SCORM are sharable content, communications, sequencing and meta-data. Sharable contents enabled through the content model, content aggregation and content packaging. Content model covers assets, sharable content objects (SCO) and aggregation.

SCO is a collection of assets, and assets are electronically representations of media, images, texts, sounds, web pages and other pieces of data that can be delivered to a web client. They both are highly reusable and in order to be reusable, assets use meta-data for search and discover purposes in online repositories [23].

SCO's may consistently be large or small depending on the specific training and education needs. They also may vary depending on the type and instructional materials that system developers create. According to the contained single learning objectives, collections of learning objectives, texts, simulations etc, the composition of the SCOs may vary too. It means different things to instructional designers and programmers. Instructional designers, authors and content developers focus on the content which is the instructional material in the SCO. For a programmer, a SCO may merely be the pointer to the actual SCO file when creating the manifest or a SCO may be a combination of SCO files and meta-data when creating the content package [23].

SCO is a collection of assets that becomes an independent, defined piece of instructional material. In an LMS they are the smallest logical unit of instruction delivered and tracked [23].

Similar to IMS, SCORM also uses content packaging. They adapt SCOs and assets into the content package while creating it. Since it is a need to exchange collections of digital resources between different LMS's, authoring tools, content repositories and operating systems', using a content package is a necessity. In traditional instructional design terms, the content package would be everything needed to deliver to the learner such as course, module or lesson.

As mentioned before, the content package contains two principal entities. The first one is the manifest file that lists all of the resources or assets you want to include in the package, the content structure diagram meaning the organization, the sequencing rules and all of the meta-data for the SCOs, the aggregations and the package itself. Aggregations are used to group the related content and defined as a parent and its children in a tree structure. In organization part, SCOs are ordered into a tree structure and sequencing behaviors are assigned to them. It outlines the entire structure created for the necessary content. The second entity of the content package is the physical files which contain all of the actual SCOs and asset files. SCOs can be viewed as any "traditional" instructional design component such as a lesson, a module, a unit, a segment or a course. As a result, it can be possible to use them in several different ways.

In SCORM, instructional designers created a tree diagram named as content structure diagram for the programmers to show the hierarchy onto which the sequencing rules for the SCOs are applied. Also in SCORM, the LMS sequences all activities between the SCOs and the learner, the designer creates some rules in order to perform all of the sequencing of the content [23].

#### 2.3.1.1.1 SCORM Meta-data

An LMS could use the meta-data to give the learner information about content aggregation (i.e. course, lesson, module etc.). Meta-data can also be used at runtime to help in the decision of what content model component to deliver to the learner.

SCORM use meta-data for tracking and locating of materials. Actually it is an additional requirement; SCORM also uses standardized file naming conventions and globally unique identifiers for tracking and locating the materials. All assets, SCOs, aggregations and content packages require meta-data in order for others to search for and locate the necessary content.

When learning resources are created, ideally the author also creates meta-data that describes the learning resource so that it can be located and reused elsewhere. Such meta-data is considered context independent of a particular collection that comprises a specific learning strategy.

The SCORM meta-data application profiles directly reference the IEEE 1484.12.1.2002 Learning Object Meta-data (LOM) standard [5] and IEEE 1484.12.3 Draft Standard for Extensible Markup Language Binding for Learning Object Meta-data Data Model [33]. IEEE provides approximately 64 meta-data elements.

The purpose of meta-data is to provide a common nomenclature enabling learning resources to be described in a common way. It can be collected in catalogs, as well as directly packaged with the learning resource it describes. Learning resources that are described with meta-data can be systematically searched for and retrieved for use and reuse.

The LOM Information Model is broken up into nine categories. These categories are based on the definitions found in the LOM information model [5].

- **general:** can be used to describe general information about the SCORM content model component as a whole.
- **lifeCycle:** can be used to describe features related to the history and current state of the SCORM content model component and those who have affected the component during its evolution.
- **meta-Metadata:** can be used to describe information about the meta-data record itself rather than the SCORM content model component that the record describes.
- **technical:** can be used to describe technical requirements and characteristics of the SCORM content model component.
- **educational:** can be used to describe the educational and pedagogic characteristics of the SCORM content model component.
- **rights:** can be used to describe the intellectual property rights and conditions of use for the SCORM content model component.
- **relation:** can be used to describe features that define the relationship between this SCORM content model component and other targeted components.
- **annotation:** can be used to provide comments on the educational use of the SCORM content model component and information on when and by whom the comments were created.

- **classification:** can be used to describe where the SCORM content model component falls within a particular classification system.

According to the IEEE, every LOM meta-data element is optional, meaning when building a XML meta-data instance, the developer can optionally pick and choose which elements to use. All meta-data instances shall have <lom> tag as the root node and it encapsulates all of the categories.

SCORM places additional requirements on which elements are mandatory in SCORM conformant meta-data XML instances. These additional requirements enable the ability to describe those objects with meta-data in a consistent manner using a consistent set of required elements; and the ability to find those learning objects in a repository so they can be used in other contexts.

## 2.4 XML, Databases and XQuery

As mentioned in the Chapter 1, in recent years two technologies are in progress for providing better performance in the usage of XML. As LOM's structure is based on XML, it is important to understand these technologies and the differences between relational databases and structured query languages.

### 2.4.1 XML

It is obvious that, the World Wide Web generally uses HTML for publishing information. Since it is an easy-to-understand language for presenting and displaying data, it has incapabilities such as presentation orientation in the means of using presentation oriented mark up tags that tell a browser how to display data to human users and having no extensibility and no data validation capabilities.

To solve these limitations of HTML, World Wide Web Consortium (W3C) offered a new standard, which aims to create a language similar to HTML in format, but more extensible and capable of clearly separating content and presentation, named Extensible Markup Language (XML) [32]. Compared to HTML, XML is also a language however, it cannot be considered as a replacement, since HTML was designed to describe data and focus on how data looks but XML was designed to describe data and focus on what data is. It was created to structure, store and to send information. By having all these properties, XML eliminates many of the limitations of HTML, especially separating content and presentation. Also, because XML is extensible, information publishers can develop their own tags. Another important property of XML is supporting validation. By associating the XML documents with a Document Type Definition (DTD) or an XML Schema, data can be validated. Since XML is based on Unicode, which includes characters from languages around the world, it is fully internationalized [32]. In the real world, computer systems and databases contains data in incompatible formats. One of the most time consuming challenges for the developers has been to exchange data between such systems over the Internet. Converting data to XML can greatly reduce this complexity and create data that can be read by many different types of applications.

Since XML data is stored in plain text format, XML has become both machine-usable and human-readable, providing software and hardware independent way of storing data. This makes it much easier to create data that different applications can work with. It also makes it easier to expand and upgrade a system to new operating systems, servers, applications and new browsers. Since XML is independent of software, hardware and application, data are available to other than only standard browsers [7].

XML is generally used by the Web and Internet users for data exchange and data integration. As XML can be used embedded within a web page with HTML, sometimes the XML data must be stored in database for further use such as sharing the content. Also, because XML is a collection of data, it seems to be a database so in order to keep these data the concept of XML databases arise.

### 2.4.2 Native XML Databases

Native XML databases are designed for creating XML data modeling [10]. For this reason, they are based on XML data modeling rather than relational data modeling. They naturally handle document order and referential integrity issues. The purpose is to provide efficiency of databases and retrieve the queries in a more rapid and effective way. Any indexes maintained by the native XML database will be directly applicable to the query language of choice and there is no translation phase when compiling a query. In an XML database, users do not have to store the physical characters associated with every tag. If the users have an XML view of the rows and tables of the relational database, it could be exactly the same size of the relational database.

Notwithstanding the current arguments, if we look at the negative side of these improvements, XML data modeling is a very new concept so it is not mature enough to handle all the features of relational data modeling. Accordingly, it is risky to use XML data model and XML database in a platform for handling meta-data.

### 2.4.3 XQuery

The other improvement in XML is the feature of XQuery. It is a SQL like language that can be used for searching both XML documents and relational databases. The World Wide Web Consortium [12] has reviewed XQuery as a formal draft [26].

Since SQL is a relational query language, XQuery is an XML query language. If all the users are querying relational databases then SQL is the answer. XQuery works best if the users querying XML or a combination of XML and relational sources. A query in XQuery is an expression that reads a sequence of XML fragments or atomic values and returns a sequence of XML fragments or atomic values [10].

XML databases running XQuery may be:

- Native: specialized engines evaluating queries on XML documents;
- Relational: built on top of existing database engines. Most commercial database products support some version of this with a subset of XQuery or XPath.

In order to use XQuery, the usage of XPath expressions must be known. XPath is used to specify structural predicate which is the most common type of predicate in XML queries. Actually it is not possible to achieve complex queries by using XQuery and also the main disadvantage is to use these queries with the help of an editor, such as Stylus Studio. It is easy to search an XML document by using XQuery, because the elements to be searched can be easily accessed over the document however, performing complex queries especially with more than one document cannot retrieve efficient answers. So, the problem is to find the best solution to keep XML data. Both relational databases and XML databases have their advantages and disadvantages according to their usage areas. Both databases have the capability of storage, query languages, schemas and programming interfaces. According to the structure of XML, storing the data on an XML database cause much more efficiency and speed compared to the relational databases. However, relational databases provide efficient storage, indexes, security, transactions, data integrity and multi-user access which cannot be used in XML databases. In order to query data, XML databases use XQuery and relational databases use SQL. Since many applications require query functionality that goes far beyond current XML Query Standards such as aggregate functions, it is obvious that SQL has a priority over XQuery. Since both databases and both query languages have different functionalities it is a must to find the appropriate solution for solving the problem of holding educational meta-data.

When we examine LOM, we found nine categories and depending on these categories there are other defined elements. If we take the first category element of LOM, named as general element, we see that there are eight main elements bound to it, as can be seen in Figure 2.12.

```
<lom>
    <general>
        <identifier>
            <catalog></catalog>
            <entry></entry>
        </identifier>
        <title></title>
        <language></language>
        <description></description>
        <keyword></keyword>
        <coverage></coverage>
        <structure>
            <source></source>
            <value></value>
        </structure>
        <aggregationLevel>
            <source></source>
            <value></value>
        </aggregationLevel>
    </general>
</lom>
```

**Figure 2.12 - General Meta-data of LOM**

As seen in from Figure 2.12 the elements bound to general element also has other elements bound to them like the *catalog* and *entry* elements of the *identifier* element. It is impossible to define such a structure in HTML because of the usage of these elements. Another reason of using XML is the tree structure. An XML document is an ordered, labeled tree in which the character data leaf nodes contain the actual data and the elements nodes are each labeled with a name and a set of attributes, each consisting of a name and value. The tree structure of the general element shown in Figure 2.13.

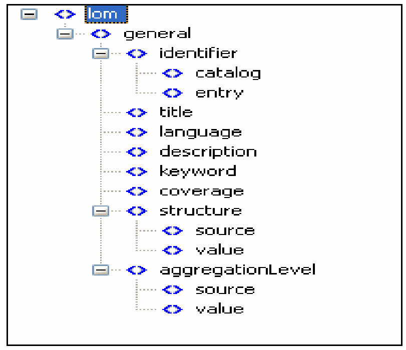


**Figure 2.13 – Tree View of General Element**

The portability of XML data and the ability of representing the data in a tree or graph format are the main advantages of XML and because LOM must be portable in order to provide sharing among different users, it seems holding these data in an XML database provide a more efficient usage. Actually, if the only problem about LOM is to store the data in a database, using an XML database will be an appropriate solution. However; storing data is not the only problem. In order to provide sharing among different users, LOM must be kept in a way to ensure complex query retrievals over multiple documents, but the incapability of XML databases via retrieving complex queries causes a confliction for the problem of LOM. From one perspective, XML databases cause efficiency for storage but from another perspective it causes inefficiency for retrieving queries over multiple documents.

The XML data of LOM is represented as a data-centric document [10]. Generally, XML documents processes over one document and by this way it become easier to retrieve queries because the fields in the document are clear. However, in LOM it is not efficient to achieve this kind of a process because LOM keeps more than one document so, this cause a problem while joining these documents with each other. Also, in order to transfer data between XML documents and XML database a mapping must be achieved.

According to the research done in this area, the problems that occur in LOM can be solved by relational databases and SQL. By using these technologies, the inefficiency and incapability of XML databases will not be an issue and by the functionality of SQL, complex queries over multiple documents can be achieved in a more efficient way.

### 2.4.4 Relational Database for XML Documents

Relational Database models can be used to create XML documents in a database. Relational databases were not designed for storing XML data so in order to keep these data in the database, XML data needs to be transformed into a relational structured form by mapping the schema of XML data to the schema in relational database. Actually, this is a long process, but it is secure and has the ability of data management capabilities such as complex queries, integrity, updates and versioning. Some of the researchers use this approach to develop a relation between XML data and relational databases [27] [30] [31].

Some of the XML data may be stored natively in XML; other data may be stored as rows and tables in relational databases. For evaluating relational systems in 1985, Dr. E. F. Codd published 13 rules that became a standard way of evaluating a relational system. Because XML is not a relational data and because XML Databases are not capable of having all the functionalities of relational databases, some of the researchers [31] do not approve hiding the information in XML as a consequence of not inconformity with these rules. On the other hand, some of them claim that XML will supersede relational data model. Also, some programmers say that XML is good only as a transient form for moving data from one place to another; others say that XML is a better data model for representing complex data structures [30].

Nevertheless, by using relational databases data will be more secure and since the LOM data will be kept as relational data in the database users do not have to deal with XML itself. Since LOM is represented with XML, there must be a connection between the data kept in the relational database and the XML view of the documents. This connection can be done by mapping the data and XML tags with each other in the developed system and automatically gives the XML conversion to the users to solve the problem of the users who do not know how to implement and use XML data. The queries will be done over the relational data and by this way; the query retrieval speed is not reduced and gives us the accurate data.

# CHAPTER 3

# MetaXML SYSTEM

In this Chapter, the software development methodology of the designed system is mentioned. After the methodology section, the metaXML System is explained with all the details.

## 3.1 Software Development Methodology

As mentioned in the Problem Statement part, for handling LOM elements and enabling meta-data sharing among different users, a new platform is required. In this platform, the aim is to provide relations among LOM elements by means of a database management system. Since it is important to understand the information domain and decide on the tools being used, a methodology must be followed.

Like in any other computer programs, it is impossible to implement the program without changing the planning, design or implementation. While progress there could be necessary changes up to the inferential requirements. For this reason, the suitable methodology is the waterfall model.

*"The waterfall model suggests a systematic, sequential approach to software development that begins at the system level and progresses through analysis, design, coding, testing and maintenance."* [28 p30].

This model derives its name from the cascading effect from one phase to another. If a problem or update occurs in any phase, a revisit required for the phases before as can be seen in Figure 3.1. The advantage is enabling testing in every phase of the model, enforcing disciplined approaches and providing documentation at every step. The classical life cycle model assumes that no defect is introduced during any development activity and from this point of view the model is idealistic. However in practice there could be defects which introduced in almost every phase of the life cycle. Therefore, we need feedback paths in the classical waterfall model. By this way it is possible to turn back to the phase in which a defect is introduced.

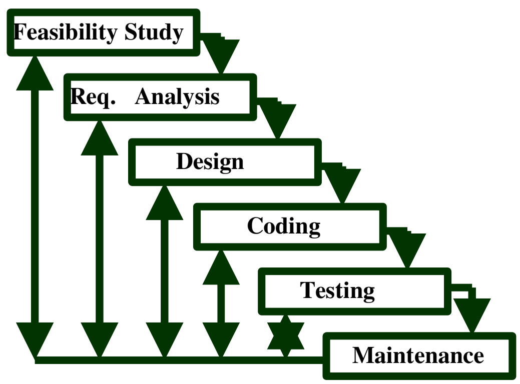


**Figure 3.1 The Waterfall Model with Feedbacks**

The waterfall model encompasses the following activities [28]:

- **System/information engineering and modeling:** *"As software is always part of a larger system, work begins by establishing requirements for all system elements and then allocating same subsets of these requirements for software. This system view is essential when software must interface with other elements such as hardware, people and databases."* [28 p31].
- **Software requirement analysis:** *"The requirements gathering process is intensified and focused specifically on software. To understand the nature of the program to be built, the information domain for the software must be understand with the required function, behavior, performance and interfacing. It is obvious that, requirements for both the system and the software should be documented and*

*reviewed with the customer(s).”* [28 p31]. Since I do not have any customer the needs are assumption based on the research done before.

In this study, system/information engineering and modeling, and software requirement analysis phases are combined and are represented as a single phase.

Because of the program is about developing an environment for educational meta-data, it is important and required to learn the meaning, purpose, usage areas and standards of meta-data itself. Frankly, meta-data is a highly complex subject. It has different meanings according to the usage areas and because of it, it is impossible to denote a specific definition.

Meta-data has a significant importance in IEEE [5] and IMS Global Consortium [4] since they both deal with educational meta-data. They define standards in order to handle the educational meta-data. After observing these standards, we decided to use IEEE standard. IEEE works with ADL initiative and ADL uses SCORM [16] to handle educational meta-data. According to the studies about SCORM, it is obvious that the structure of the learning educational meta-data has the best structure for the platform we have designed. The categories and the elements bound to these categories are examined and their meanings, data types and multiplicities are deducted as a summary.

On the side of these analyses, another important research must be done in the meaning of understanding the other studies about educational meta-data. There could be so many different approaches while designing such a system so, in order to get a second opinion about the design and implementation, these approaches are examined. In respect of these studies, we breakdown the insufficient parts and tried to use them as a plus in this study.

- **Design:** **“***The design process translates requirements into a representation of the software that can be assessed for quality before code generation begins. Like requirements, the design is documented and becomes part of the software configuration”* [28 p31].

After analysis part, the design process begun, according to the requirements of the platform. The aim is to translate the system specifications into a software representation. The first thing is to develop the relational data based on the SCORM's learning educational meta-data. As mentioned before these data are kept as XML and for the reasons given in the Chapter 2, it is not reasonable to use an XML database, so it is a must to design the entity relationships of these elements in order to use them in the relational database. *"An entity relationship model is a high-level conceptual model that describes data as entities, attributes and relationships."* [29 p62]. In order to show the representation of data, the entity relationship model is supported with E-R diagrams. The purpose of these diagrams is to enable designers and users to put through what the planned database intend to do or how it could work, not to specify the actual data or store it. Data storage is the second step depending on the data content and the database management system.

While the deduction of drawing E-R schemas, there are some certain considerations to be followed based on the SCORM learning object meta-data. These considerations are parent-child relationships between the elements, package multiplicity and data types of the elements. The relationship between the parent-child elements of the categories are designed by the help of XML notifications of the elements and by the help of the studies of SCORM. The multiplicities of the elements are also based on SCORM. However, in SCORM there are five different application profiles named as package, content aggregation, activity, SCO and asset. Application profiles' main purpose is to provide modularity and extensibility [35]. The multiplicities of the elements differ based on these profiles. According to the profiles some of the meta-data elements can be generated automatically however, a significant number of elements should be inserted by the user and this causes an amount of time for the elements which can occur for example 30 times [36]. In the schemas, package application profile is used since it covers all of the other profiles. The basic difference between package and the other application profiles is about participation. Generally, all elements are optional according to the package application profile.

The other consideration is the data types of the elements. There are some specialized data types as vocabulary and LangString. Both types can be used by other elements so they are designed as separate entities. The elements not using these data types are generally represented with CharacterString data type. The important issue about this data type is the language. If there is a string then the language code must be used for helping the users to understand the data's represented language(s).

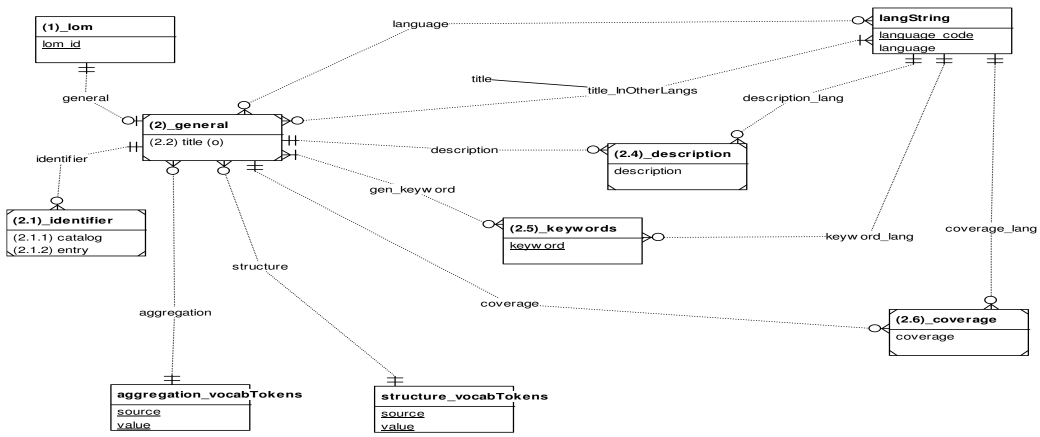


**Figure 3.2 – General Element's E-R Schema**

The Figure 3.2 represents the E-R schema of the general element. Like the other category elements, it is derived from <lom> element. By the help of multiplicity of elements cardinality and participation properties are given. Some of the elements can occur more than once according to the smallest permitted maximum property. The smallest permitted maximum indicates that applications that process meta-data shall process at least that number of elements or characters but are free to support and exceed the limit. This property shows us that the elements can be flexible according to the needs of the users while developing the programs about educational meta-data.

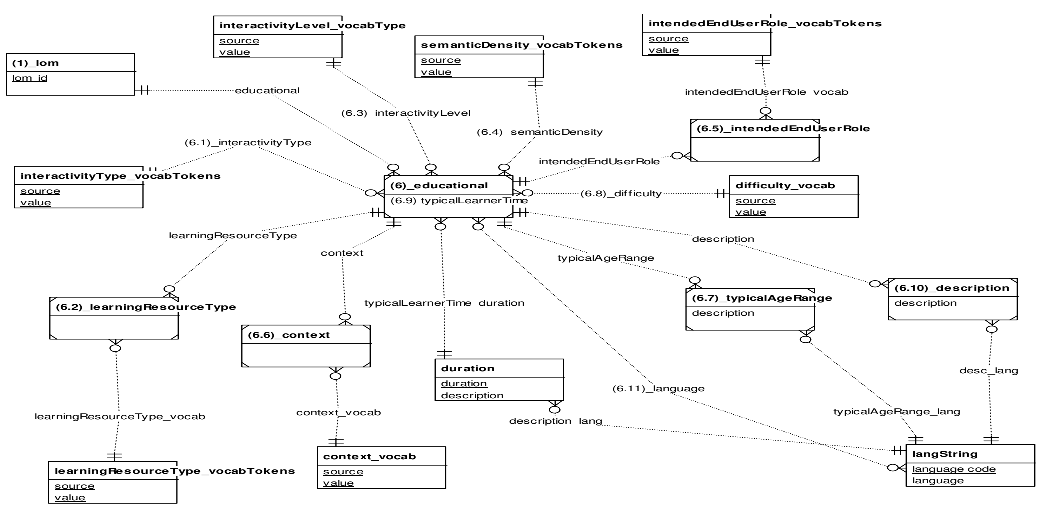


**Figure 3.3 – Educational Element's E-R Schema**

Figure 3.3 also represents an E-R schema of another category named educational. It is also derived from <lom> element. Educational category has so many different elements and most of them are using vocabulary data type, so designing the schema is harder than the other category elements.

After designing the schemas of nine category elements, the tables and attributes are determined for database storage.

- **Code generation:** *"The design must be translated into a machine readable form. The code generation step performs this task."* [28 p.31]. Implementation phase is the most challenging part since according to the needs of coding, the design might be changed.

In this phase, the design of the system is translated into the software domain. To reduce coding effort, detailed documentation must be done in the design phase.

The platform must be web based because of the aim is to provide sharing among different users. This could be done by only using the right web tools. There are so many options like asp, php or java but we preferred to use php. It is an open source server-side scripting language for creating web pages for web based applications. There are so many advantages of php such as; embedding php codes inside html, similar syntax of C language and ease to maintain and update than comparable scripts in other language. Since it is an open source language, the community gives excellent technical support to their users. Also the code is continuously updated with improvements and language extensions to expand php's capabilities. Another advantage of php is the database connectivity. It offers connectivity to most of the common databases including Oracle, Sybase and MySQL. It also offers integration with various external libraries, which allow the developer to do anything from generating PDF documents to parsing XML.

There are three main areas where php scripts are used. The first one is the server side scripting which is the most traditional and main target field for php. The second one is command line scripting which enable users to make a php script to run without any server or browser. The only thing needed is a php parser. The third one is writing desktop applications.

Php can be used on all major operating systems, including Linux, Microsoft Windows and many UNIX variants. Also supports for most of the web servers including Apache, Microsoft Internet Information Server, Netscape and many others. So with php, it is possible to choose any operating system and web server supporting with choosing of using procedural programming or object oriented programming or the mixture of them.

For database connectivity we prefer MySQL in consequence of becoming the world's most popular open source database according to its consistent fast performance, high reliability and ease of use. For programmers it is important to have fast database connectivity while retrieving queries. There could be so many tables and data so; no one wants to wait for a long time for seeing the results of the queries. MySQL gives high performance whether the intended application is a high speed transactional processing system or high volume web site that services a million queries a day by using high speed load utilities, distinctive memory caches and other performance enhancing mechanisms.

MySQL provides strong data protection based on the database authentication. The system allows for only authorized users to enter to the database server with the ability to block users down to the client machine level. Users only see the data they should, and data encryption and decryption functions ensure that data is protected from unauthorized viewing with the help of object privilege framework.

While using MySQL, an interface can be used for creating tables, data insertion, deletion or creating queries. The interface is phpMyAdmin, a php application running on a php enabled web server. The aim is to handle the administration of MySQL over the web.

By using php and MySQL, the implementation phase is easy to maintain comparing to other languages. There are some changes reflexes especially to the E-R schemas of the learning object meta-data elements and tables during the implementation and testing processes.

Testing has an important place during implementation. It focuses on making sure that any errors are identified and that the software meets its required specification.

- **Testing:** In this phase system is tested. Normally programs are written as a series of modules and these modules are tested separately. After the modules are completed, they brought together and tested as a complete system.

The system is divided into four modules as data input and output, inside search, search in all elements and converting data into XML. All modules are tested separately during the implementation phase. Then the system is tested to ensure that the interfaces between modules work (integration testing), the system works on the intended platform and with the expected volume of data (volume testing) and that the system does what the user requires (alpha testing).

The reason of using integration testing is to understand if these four modules can work together as a whole to verify functional, performance and reliability requirements placed on major design items. During integration testing bottom up method is followed because of giving the capability of testing individual modules from a test harness. Once a set of individual modules have been tested they are then combined into a collection of modules, which are then tested by a second test harness. This process can continue until the build consists of the entire application. As an example, the module data input and output has nine different modules because of implementing nine different category elements. Starting from the first element, each category tested individually and then combined together and tested as a whole module. Since search modules and converting into XML modules depends on data input and output, it is a must to fix all the problems caused from coding or database. After testing of data input and output module, converting into XML module is implemented and tested, and lastly search modules are implemented and tested. The design of the system changed according to the problems faced during implementation and testing these modules.

After integration testing, to understand the system's efficiency, volume testing is applied. The purpose is to check if there are any problems when running the system under test with realistic amounts of data. The test data are taken from IEEE LOM [5] and IMS Global Consortium [4]. Especially using databases in a system can cause problems like data loss or redundancy of data. Also the elements use different data types like date and language and some of the elements are required to be inputted by the user. During volume testing, actual data can give information about how data must be entered. Also it is important to check error messages and warnings while entering data and get the outputs from the database.

The last phase of testing is done by using alpha testing. A couple of programmers test the system and in respect of their warnings the system converted into a more user friendly platform.

### 3.2 MetaXML System

The purpose of developing an environment for educational meta-data is mainly providing the users for meta-data sharing between them and making logical queries using the relational database. After examining the research done before, it is obvious that there is a need of a new system covering these needs. For this reason metaXML is developed.


**Figure 3.4 – Entrance Page of MetaXML System**

Figure 3.4 represents the opening page of the program. When the user(s) enter to the web site, a short brief about the program appears in the opening page. If the user is new to the program then he/she must be registered to get a user id and a password. If the user(s) used the program before, then they could login by using their usernames and passwords. In metaXML users are defined as web users because we want them to access to the whole system especially for the search purposes. The users are kept in the database for future entering, but if the users forget their usernames and passwords, the system provides them to take new passwords or usernames, since the data they entered are not kept as hidden data.

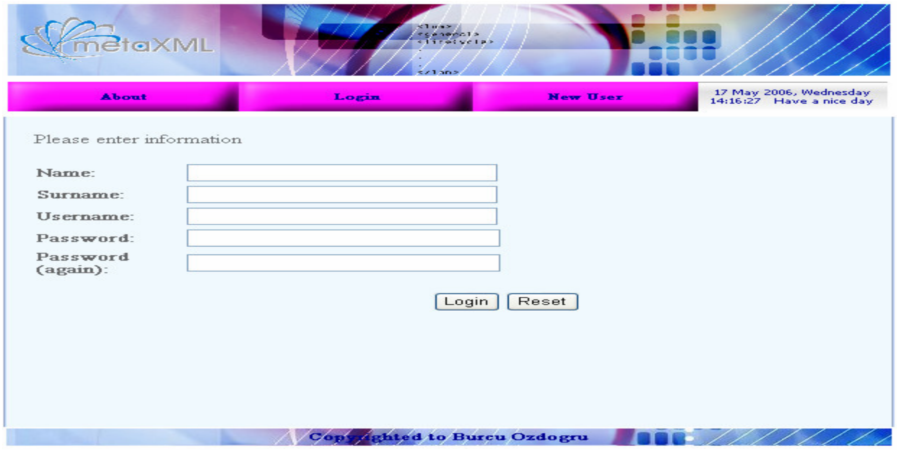


**Figure 3.5 – New User Page**

If the user is new to the system by using "New User" page, he/she can enter the required information and registered to the system as can be seen in Figure 3.5. While entering the information, it is important to check the data for accuracy. Also it is required to fill all the fields so if the user forget to fill any of the required information, then the program gives warning until all the fields are filled

The program also checks if the entered username exists in the database or not. If the username is found in the database then another warning is given to the user for changing the username.

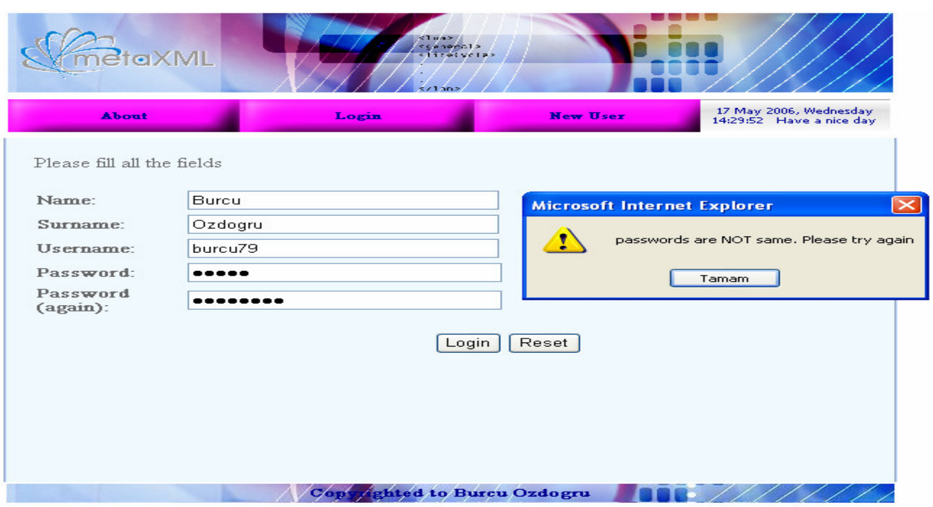


**Figure 3.6 – Password Check**

Like in any other web based programs, password fields are very important. In metaXML, password fields are supported with double checking. If the user does not enter the same password to the related fields, then a warning message appears as can be seen in Figure 3.6.

The required data is kept in the database in a table named "new_user". Since the user(s) could have the same name and surname, an attribute named "user_id" is defined for primary key purposes. After cleaning out the errors (if any) and entering the accurate data, the user enters the "Login" button and the data is sent to the "new_user" table, as can be seen in Figure 3.7


| Alan | Tip | Karşılaştırma(Collation) | Özellikler | Boş | Varsayılan | Ekstra | Eylem |
|---|---|---|---|---|---|---|---|
| user_id | int(11) | | | Hayır | | auto_increment | |
| name | varchar(30) | utf8_general_ci | | Hayır | | | |
| surname | varchar(30) | utf8_general_ci | | Hayır | | | |
| username | varchar(20) | utf8_general_ci | | Hayır | | | |
| password | varchar(20) | utf8_general_ci | | Hayır | | | |

**Figure 3.7 – New_User Table**

If user is already defined before, then he/she use the "Login" page for entering to the system.

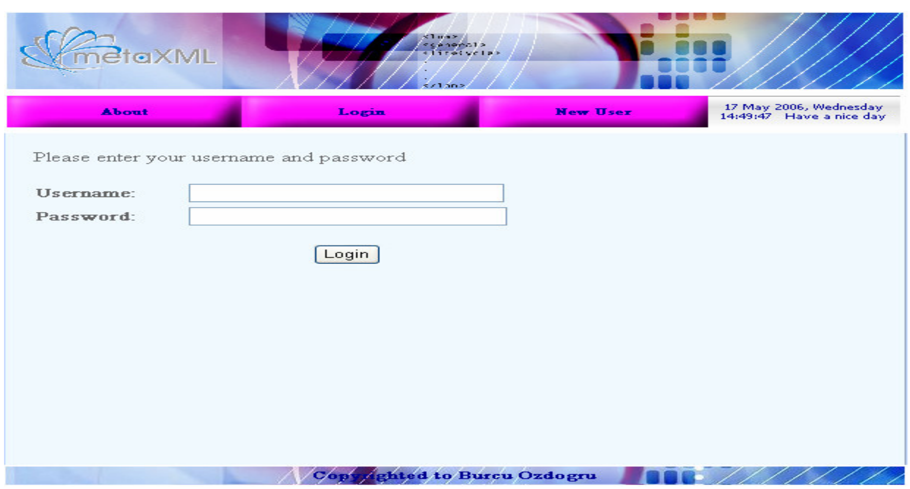


**Figure 3.8 – Login Page for Existing Users**

As can be seen in the Figure 3.8, there are also checks related to the database. The username and password must match with the data in the "new_user" table. For example, if the user could not be found in the database, then a warning message occurs. Also, another warning message occurs if the username is correct but password is incorrect.

After entering the correct information the user enters to the system.


**Figure 3.9 – MetaXML System**

The Figure 3.9 represents the opening page of metaXML program after login. There are three different links named "How to…", "New LOM Document" and "Search in ALL". These three links have different aims. "How to…" link gives an opinion to the users for using the program as can be seen in Figure 3.10.


**Figure 3.10 – How to... Page**

If the users want to enter meta-data then, they have to choose "New LOM Document". This link opens with general element's form. There are nine different elements in learning object meta-data and it is possible to begin with any of these elements, but since the users are defined as web users it is not logical to give them the privilege of using the which ever elements they like to use. For this reason, users can reach the other elements by using the relevant buttons.

In metaXML, search mechanism is used for two purposes. The first one is enabling users to search in the whole database meaning all the category elements and generating more complex queries. The second one is enabling users to do search bounded to the categories. The reason for using this kind of search is to give the opportunity to the users understanding of what kind of data are used while entering the information.

The elements are organized by the help of SCORM's XML notifications. According to the smallest permitted maximum property, some elements could be occur more than once, generally 10 times. In metaXML, the elements which have smallest permitted maximum are arranged as to appear only three times.

Since these elements are kept in the database it is important to decompose them for preventing data redundancy. For this reason, all elements have a common lom_id. While creating forms, xml notifications and doing search by using these elements, lom_id become a uniting element.

As mentioned in Section 3.1, there are four modules in metaXML, mainly data input/output, xml conversion, search in elements individually and search in all elements. While explaining these modules, XML conversions included into the other modules because user(s) can examine XML notifications of the LOM elements in every other module.

### 3.2.1 Data input/output Module

MetaXML is designed for providing users to do the data input by themselves, because of not having countless of test data about learning educational meta-data. Also, the existing test-data languages are mainly in English.

Reaching test-data about LOM is possible through IEEE [5] and IMS [1] but the language of existing test-data is mainly in English. In order to have accurate data about LOM in Turkish users must enter their own data to provide a pool of data about LOM. Furthermore, the main purpose of designing such a system is to provide sharing among users so, entering accurate data will help users especially while searching in the LOM.

LOM has nine elements and these elements are differentiated based on their categories and offered to the users in this manner. Since there is no required order between LOM elements, users have the ability of entering data by using any of the categories. However, since LOM gives designers an option of implementing a system as their own request, in metaXML we preferred to put the elements in an order. The tables keeping the entered data have a primary key named "lom_id" and with this key data are differentiated from each other preventing clutter between them.

In metaXML, general category element is chosen as the starting point of the program, because this element groups the general information that describes the resource as a whole it can be viewed as key information for describing the particular content.

Users do not have to enter data to general category elements (it is not required), it is enough to select the "add" button in the form for creating the lom_id, and after this creation the id is sent to the next category element. For the following eight category elements, the program gives users two options. Users can enter data to the related category, after entering them by pressing the "add" button they can reach to the next category element. If users do not have any data related with the category element, by pressing the "skip" button, they can reach to the next category element. In either way the "lom_id" is created in the general category form and sent to the following categories.

While entering data into the categories, a set of controls should be made because of preventing data redundancies and data anomalies related with the database. These controls are based on; firstly the elements which may repeat three times in the related forms, secondly the elements which are represented by LangString data type and thirdly the elements which are required to be entered by the user.

In LOM, only the "metadataSchema" element in the "metaMetadata" category element is required, the other elements are optional, so the control mechanism is based on this situation.

For the elements which may repeat three times in the related forms, the system blocks the users by preventing from entering the same data to the repeated fields. For the elements defined by using LangString data type, the languages are given to the users by the help of a list box and the system forces them to enter a language to the related fields.

Another situation is about the elements which may repeat three times with an option of having three different languages. For situations like this, control starts from the first group of data and users are forced to enter different languages for the repeated fields, if any. Control continues by checking the data groups with each other and entering same data for two or three different groups are prevented.

For all the controls done in the category elements users get warning messages about the errors they make while entering the data. These warning messages are connected to the "add" button which exists in all the data entry forms. By this method, the errors are corrected before the data sent to the database to the related tables.

After entering data and pressing "add" button, another page is opened providing the users to see what they entered to the related category element. In this page users may examine the xml notification of the data they entered to that category element by using "xml_format" button. The converted data can then be used in an xml editor according to the needs of the users.

For reaching the next category element, users use the button indicating the name of the following category.

In the last category element named classification, users may see either the data they entered in the classification form as xml or all the data they entered for that learning object meta-data component as xml. By this way, they can examine and use the xml notification of the regarding component as a whole.

If users want to define a new LOM component then they have to go back to the main menu of the program and select the "New LOM Document" link. The process continues in the same way after selecting a new document.

To explain the system's operations in a more detailed way, general element is used as a demonstration.

### 3.2.1.1 General Element

When the users selects "New LOM Document" link, the first form that appears in the system is the general category form, which can be examined by the help of Figure 3.11. The elements defined in the form are the child elements of general category and appear in an order, which is based on SCORM's LOM. Consequently, the xml notifications of these data are created in this order too.

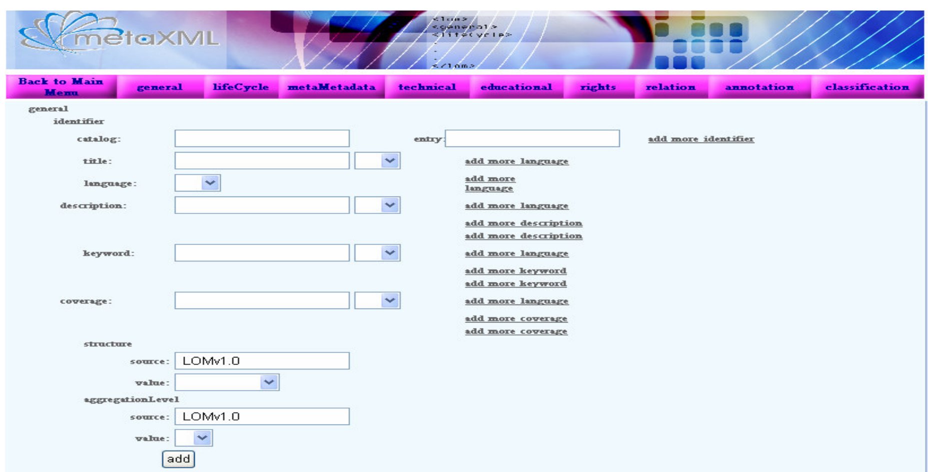


**Figure 3.11 – General Category Page**

Some of the elements are repeatable and to differentiate them with non-repeatable elements, the system embed java script buttons named as either "add more language" or "add more <element_name>" to the form. By clicking these buttons, the second field of the repeatable element appears in the form, and by clicking again the field will disappear. By the help of these scripts, the program provides a more user-friendly platform.

As an example, identifier element is repeatable and has a structure containing of these java script buttons, as can be seen in Figure 3.12.

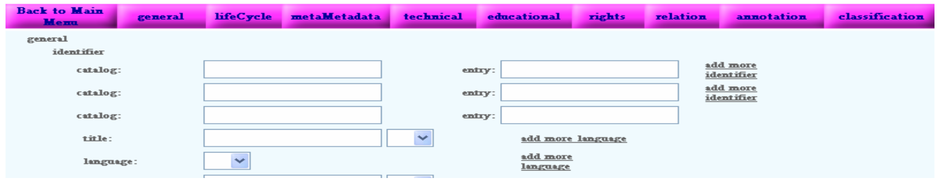


**Figure 3.12 – Identifier Element**

The description, coverage and keyword elements' situation differs from the other elements since they have three repeatable fields with the representation of three different language options for all of them separately. For this reason the scripts "add more language" and "add more <element_name>" used together in these fields. This structure can be examined by the help of Figure 3.13, which represents the description element.

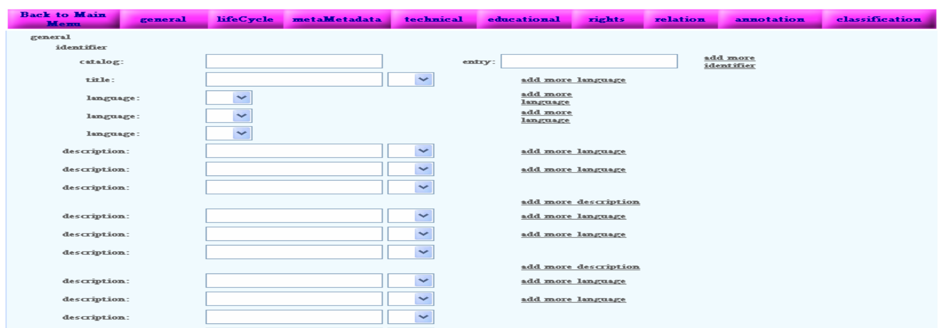


**Figure 3.13 – Description Element**

The elements defined by using LangString data type such as the title element, as can be seen in Figure 3.14. Languages appeared by the help of a list box. These languages are kept in the database in the related table named LangString, enabling data transactions such as update and delete to provide efficiency. So, if a change is needed, only the related table is updated and the changes are reflected directly to the program. This situation also valid for the elements defined with vocabulary data type. However, vocabulary data type is different from LangString data type because in LOM, every element represented with vocabulary type has different vocabulary tokens, so these tokens are kept in the different tables. As an example, Figure 3.15 and Figure 3.16 can be examined showing the elements structure and aggregationLevel which are defined with vocabulary data type.

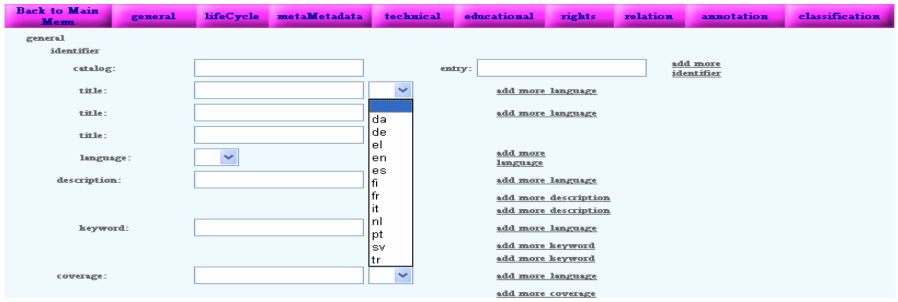


**Figure 3.14 – Language Representation of Title Element**

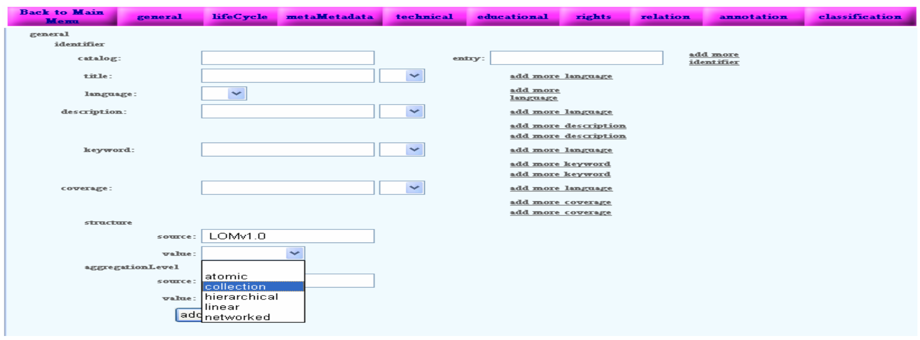


**Figure 3.15 – Structure Elements' Vocabulary Tokens**


**Figure 3.16 – AggregationLevel Elements' Vocabulary Tokens**

### 3.2.1.1.1 Data Control

While enabling users to enter data by using forms, the entered data are sent to the related tables of the elements when the user presses the "add" button. Before sending these data, the program checks the entire form for preventing the errors caused by not entering the language fields of the elements or data redundancy for the elements which could appear more than once. Since all the elements in the general form are optional, there is no check mechanism for empty fields.

The checking mechanism starts with identifier category. The user(s) could enter three different data of catalog and entry, so the program checks whether the entered data are the same data for the three of them. If the entered data is the same the program gives a warning message to the user for changing the data.

The mechanism continues with the title element. Here the program checks two things. The first one is the title-language relationship. Since the title could not exist without the language, the program checks whether the user forgot to enter the language of the title element. The second check is about giving the users the opportunity of entering three different languages of the title element. Here the titles could be the same since in some languages, the writings could be similar but the users must choose different languages. So, the program checks if the entered languages are the same.

After title element, the program checks the language element since it is possible to define more than one language for SCORM, so for this element the program looks whether the entered languages are same or not. If they are same, a warning message appears providing users to change the entered data.

The elements description, keyword and coverage have the same checking mechanism differing from the other elements of the general element. These elements could appear three times with three different language options. The program first checks the elements according to the language then checks as a group to understand whether the user enters the same elements or not. As an example consider the keyword element. For the keyword meta-data, user wants to define three different languages. Here the program checks the languages; if user enters the same languages for the keyword meta-data then a warning message appears. If the user wants to define another keyword, then the program checks whether the entered keyword is same with the other keyword which is entered as meta-data or not. If user enters meta-data for the second keyword group then the program gives another warning message for enabling to change the keyword.

#### 3.2.1.1.2 Database – Form Relationship of General Element

After checking the errors, all the data entered are sent to the database to the related tables. The tables of general category element can be seen in the following figures, beginning with Figure 3.17.

| Alan | Tip | Karşılaştırma(Collation) | Özellikler | Boş | Varsayılan | Ekstra | Eylem |
|---|---|---|---|---|---|---|---|
| lom_id | int(11) | | | Hayır | | auto_increment | |
| title | varchar(255) | utf8_general_ci | | Hayır | | | |
| language_code | varchar(8) | utf8_general_ci | | Hayır | | | |
| structure_source | varchar(8) | utf8_general_ci | | Hayır | | | |
| structure_value | varchar(20) | utf8_general_ci | | Hayır | | | |
| aggregation_source | varchar(8) | utf8_general_ci | | Hayır | | | |
| aggregation_value | varchar(2) | utf8_general_ci | | Hayır | | | |

**Figure 3.17 – General Table**

Since the title, structure and aggregationLevel elements occurs 0 or 1 time in the program, they are created in the general table. For title element, only the first entrance of data is kept here, the other language options for title element is kept in another table named as titleInOtherLanguages which is shown in Figure 3.18.

| Alan | Tip | Karşılaştırma(Collation) | Özellikler | Boş | Varsayılan | Ekstra | Eylem |
|---|---|---|---|---|---|---|---|
| lom_id | int(11) | | | Hayır | 0 | | |
| language_code | varchar(8) | utf8_general_ci | | Hayır | | | |
| title | varchar(255) | utf8_general_ci | | Evet | *NULL* | | |

**Figure 3.18 – TitleInOtherElements Table**

The structure and aggregationLevel elements' data type is vocabulary and they to have to use the defined vocabulary tokens. These tokens are also kept in the database for future changes of these vocabulary tokens such as adding or deleting a token.

| Alan | Tip | Karşılaştırma(Collation) | Özellikler | Boş | Varsayılan | Ekstra | Eylem |
|---|---|---|---|---|---|---|---|
| lom_id | int(11) | | | Hayır | 0 | | |
| language | varchar(8) | utf8_general_ci | | Hayır | | | |

**Figure 3.19 – Language Table**

In language table as can be seen in Figure 3.19, the language element is kept. Since it is possible to define three different languages, the differentiation is done by using lom_id. As an example, assume the lom_id is 23 for a bunch of learning object meta-data and if the user enters three different languages, these are kept in the database like 23 da, 23 en, 23 nl which represents the lom_id and language_code respectively.

Description, coverage and keyword tables are similar since they have the similar properties. In these tables data are differentiated by using lom_id and language_code since it is possible to have three different data with three different language codes. Also all of the three attributes in each table are defines as primary keys to prevent wrong data entrances. The table descriptions of these elements can be seen in Figure 3.20, Figure 3.21 and Figure 3.22.

| | Alan | Tip | Karşılaştırma(Collation) | Özellikler | Boş | Varsayılan | Ekstra | Eylem |
|---|---|---|---|---|---|---|---|---|
| | lom_id | int(11) | | | Hayır | 0 | | |
| | description | varchar(255) | utf8_general_ci | | Hayır | | | |
| | language_code | varchar(6) | utf8_general_ci | | Hayır | | | |

**Figure 3.20 – Description Table**

| | Alan | Tip | Karşılaştırma(Collation) | Özellikler | Boş | Varsayılan | Ekstra | Eylem |
|---|---|---|---|---|---|---|---|---|
| | lom_id | int(11) | | | Hayır | 0 | | |
| | coverage | varchar(255) | utf8_general_ci | | Hayır | | | |
| | language_code | varchar(6) | utf8_general_ci | | Hayır | | | |

**Figure 3.21 – Coverage Table**

| | Alan | Tip | Karşılaştırma(Collation) | Özellikler | Boş | Varsayılan | Ekstra | Eylem |
|---|---|---|---|---|---|---|---|---|
| | keyword | varchar(255) | utf8_general_ci | | Hayır | | | |
| | language_code | varchar(6) | utf8_general_ci | | Hayır | | | |
| | lom_id | int(10) | | UNSIGNED | Hayır | 0 | | |

**Figure 3.22 – Keyword Table**

#### 3.2.1.1.3 Form and XML Notification of General Element

The page providing users to see the data in a form view depends on the “add” button of the data input page. By this button, first the data controls done, secondly the data send to the relevant tables and third by using the tables necessary data are taken and put into the form view.

The data input form is seen in from Figure 3.23

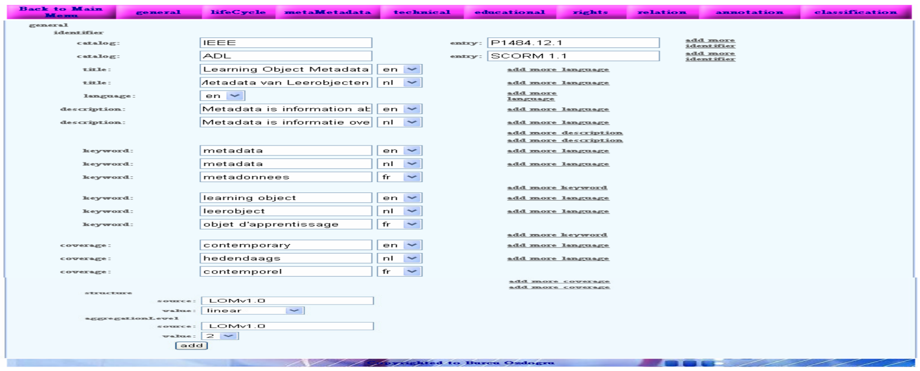


**Figure 3.23 – Input Data of General Category**

After sending these data to the database, another page is opened providing users to see what they entered as seen in Figure 3.24.


**Figure 3.24 – Output Page Referring to the Data Input**

In this page user(s) have two options, one is to see the xml format of the data they entered, and the other one is a link to the lifeCycle element. If the user selects the "xml format" link, the xml version of the data opens in a blank page, as can seen in Figure 3.25.

```xml
<lom>
<general>
   <identifier>
      <catalog>ADL</catalog>
      <entry>SCORM 1.1</entry>
   </identifier>
   <identifier>
      <catalog>IEEE</catalog>
      <entry>P1484.12.1</entry>
   </identifier>
   <title>
      <string language ="en" >Draft Standard for Learning Object Metadata</string>
      <string language ="nl" >Voorstel van Standaard voor Metadata van Leerobjecten</string>
   </title>
   <language>en</language>
   <description>
      <string language ="nl" >Metadata is informatie over een object. Het kan hierbij gaan omeen fysisch of een digitaal object.</string>
      <string language ="en" >Metadata is information about an object, be it physical or digital.</string>
   </description>
   <keyword>
      <string language ="en" >learning object</string>
      <string language ="nl" >leerobject</string>
      <string language ="en" >metadata</string>
      <string language ="nl" >metadata</string>
      <string language ="fr" >metadonnees</string>
      <string language ="fr" >objet d'apprentissage</string>
   </keyword>
   <coverage>
      <string language ="en" >contemporary</string>
      <string language ="fr" >contemporel</string>
      <string language ="nl" >hedendaags</string>
   </coverage>
   <structure>
      <source> LOMv1.0</source>
      <value>linear</value>
   </structure>
   <aggregationlevel>
   <source> LOMv1.0</source>
   <value>2</value>
  </aggregationlevel>
</general>

</lom>
```

**Figure 3.25 – XML Representation of General Category**

The XML conversion is done by using the data which is kept in the database. By the help of lom_id, all the tables related with the general element is searched and called by the program. Since the data retrieval is easy by using php-mySQL relationship, this method is very efficient. After converting the XML notification of the elements, it is possible to use them in an XML editor. While testing the program, the validation of XML elements is also checked by using XMLwriter program.

After seeing the XML version of the entered data, the user could continue to enter data by using the next element named as lifeCycle.

### 3.2.2 Search Modules

Depending on the reasons explained in Chapter 2, xml is not adequate for retrieving queries and the method XQuery cannot fully answer the problems of xml, so in order to provide meta-data sharing between users, search module has an important aspect in metaXML.

Programs designed for using LOM generally provide data input and xml conversions however, these programs do not offer complex search options. The main reason is the will of keeping the data as xml. Because if data are kept as xml the conversions will be much more easily done, but at that time the efficiency of using complex queries decreases depending on this situation. It is obvious that, converting the data to xml is not enough while dealing with LOM, so the designed programs must provide complex query retrievals. Most of the software companies offer these kinds of programs but since they are not freeware, most of the educators can not take advantage of them. Whereas, if these kinds of programs are offered to the educational world, most of the people will use and benefit from sharing the learning object meta-data components.

Therefore, metaXML is developed for providing search capabilities to help the educational world to get what they are seeking for. The search options are designed for two purposes. The first one is designed for the users who want to do data input through the system, and the second one is designed for the users who want to search the whole system by using complex queries.

#### 3.2.2.1 Search in Categories Individually

In the general sense, the users who want to enter data to the system, wants to see the previous data entered for that category element. In metaXML, by the help of search buttons, users have the ability to do search in any category element they want. In this search module, users can search inside the particular category element by using the necessary search fields and outgoing results are shown by using the relevant tables of that category element. However, if users do not have different learning object meta-data other than the information they have, they can pass to enter data into the search fields and just press "search" button. Users are not required to fill the search fields, they can use the related fields or all of the fields while doing search.

The fields providing search are chosen by examining the test data of IEEE [5] and IMS LOM [1]. According to the analysis, it is seen that some of the data are not used and some of them are used frequently. So, the data which is used more is considered and presented as search fields in the program.

After searching the database, the results are sent to a table and this table appears in the related search form. To differentiate the coming results, metaXML uses the search fields as column titles so the results are sent to the relevant columns' attributes. Because the search is done by using some of the elements of the category element, it is a must to give users a chance to observe the whole data in that category.

Since it is not a right approach of restricting the users to see the results as form or xml data, metaXML gives users these two options both so; the users may observe the results by choosing any of the options. For these reasons, after the results appear in the table, users can see the form or xml notification of the related category element by using "form" and "xml" buttons which appears near to all result attributes. If users select the form button, all the data of the related result will shown in a separate form page. If users selects the xml button, then all the data related to the selected result will shown as xml data in a separate page.

Another reason of showing the results by using xml notification is to help the users who do not know how to implement the data in xml. Also, users may use the resulting xml notification for their needs.

Users have the ability of observing any result they want. If the system cannot find any record in the database related to the data entered into the search fields, then only the columns of the table appears in the form with the search fields. So, users may enter another search data and wait for the result(s).

To explain the search module's operations in a more detailed way, general element is used as a demonstration.

#### 3.2.2.1.1 Search in General Element

In general element, search can be done by using title, keyword, coverage and/or structure elements as seen in Figure 3.26.

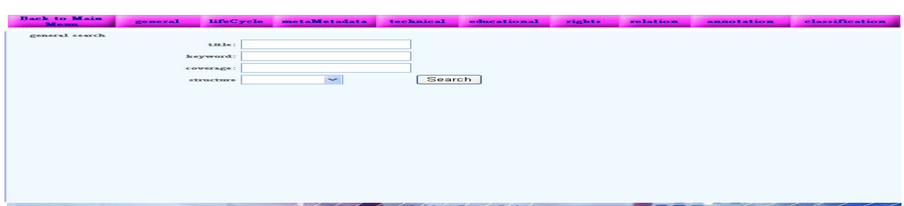


**Figure 3.26 – General Search Fields**

While searching, due to the users who do not have any idea what to enter, the search fields can be null. If the search button is clicked while all the fields are empty, then the system calls all the general elements from the relevant tables where the results can be examined by the help of Figure 3.27.


**Figure 3.27 – General Search Results**

As can be seen in the Figure 3.27, there are eight entrances of data for the general element. The data entered as an example in the data input/output module in general category [page..], appears as the eighth result in the table. Continuing the following steps by using this result, the form view and xml notification of the result may be examined in the Figure 3.28 and Figure 3.29.


**Figure 3.28 – Form View of the Selected Result**

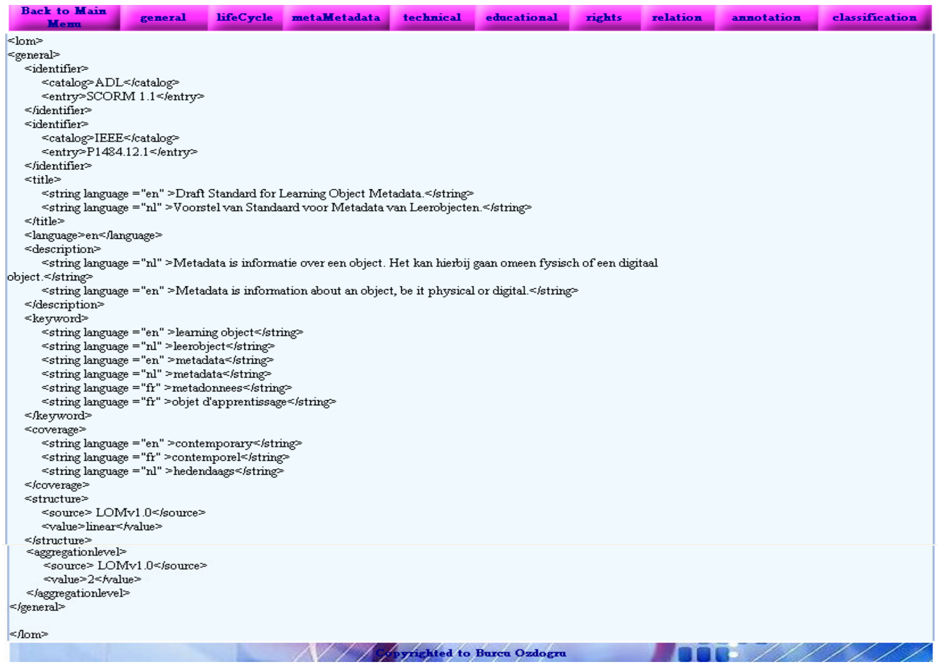


**Figure 3.29 – XML View of the Selected Result**

If the user(s) fill the search fields, system searches relevant tables and output the results. According to their demand, users may do search either in all the fields or individual fields. In Figure 3.30, user fills the keyword and structure fields so the search is done for these two data. If the system finds a record containing both of the desirable data, then the result appears inside the table.


**Figure 3.30 – Search Result Based on the Selected Search Fields**

If the system does not find any record than the output fields appear as empty as can be seen in Figure 3.31.

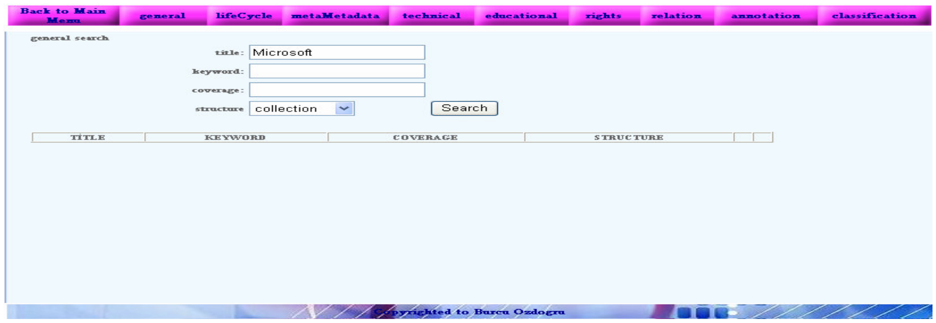


**Figure 3.31 – No Results Page**

### 3.2.2.2 Search in All Categories

The second search module provides users to search in the whole database by the help of necessary search fields.

The reason of designing such a search module is the deficiency of the other search module, since that module is designed only for the specified categories. However, in order to provide users to make complex searches, the system has to enable them to examine the whole learning object meta-data structure for the necessary components. Therefore, this module allows users to search the whole database by using the chosen search fields. The process is same with the other search module but from the programming side there are much complex queries done. Especially because of using nine of the learning object meta-data elements while retrieving queries, it is important to get the best efficiency from the users' point of view, so the system is designed to be as fast as possible while retrieving the queries.

In this module, metaXML give users two options. The first one is the option of searching the whole database by using the category elements individually by using a list box covering all the category elements. When user picks an element from the list box, the search fields of the related category element will appear. The second option is the ability of searching the whole database by using the chosen fields of nine category elements. In this option the search fields are decided according to the users' needs. The results of the queries are kept in a table and for differentiation of the results, the column names are used. Because users need a more meaningful column name to understand the results, system uses the fields of title, keyword and format. As in the other search module, users are not required to fill all the fields. When the "search" button is clicked, the whole database is scanned and the results appear in the table. In both of the search options in this module, the results can be viewed by using form or xml. Nevertheless, the views are showing the elements of the nine category elements in one page in the order of inputting data module. Consequently, especially with the help of the title column in the results table, users can have the ability of examining the whole xml notification or the whole the form view of the component they are interested in.

As an illustration, Figure 3.32 shows the search modules form.

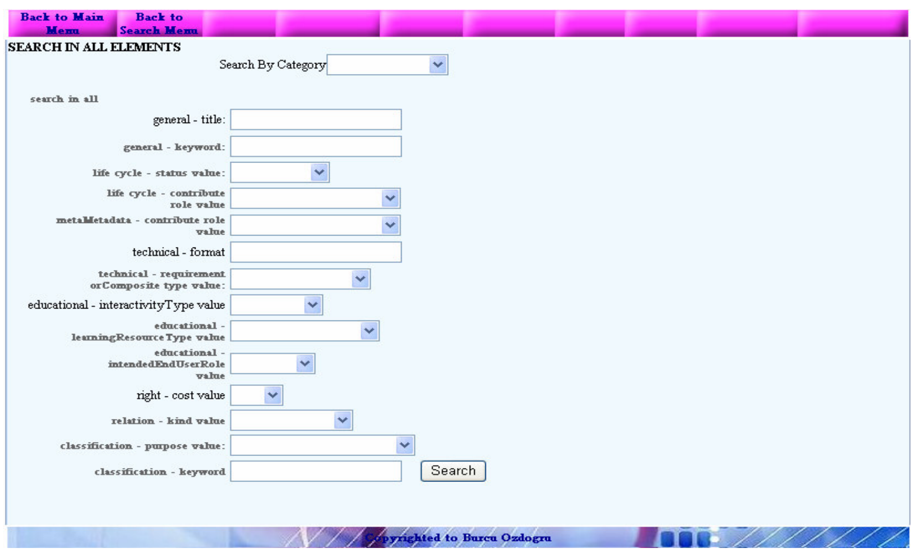


**Figure 3.32 – Search in All Elements Page**

To search the category elements individually, users select the category they are interested in from the “Search By Category” list box. According to the selected category element, necessary search fields appeared in a new web page.

To explain this option in a more detailed way, educational element is used as a demonstration.

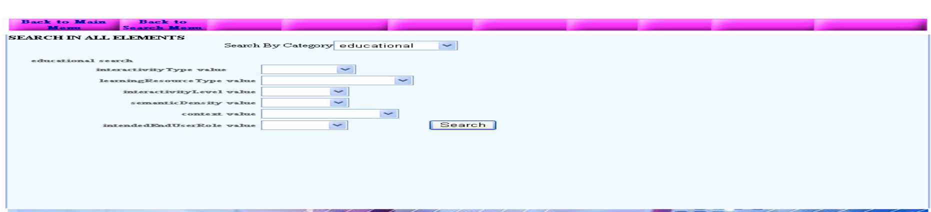


**Figure 3.33 – Educational Categories’ Search Fields**

Figure 3.33 shows the educational elements' search fields. The search fields are the same with the fields of individual search module. They both search only the selected category but there is a difference as in this search users can see the components with all the properties defined by learning object meta-data elements. This can be done with the help of "lom_id". When users search in the educational category for a component, with "lom_id" the tables are linked together and by this way the result of that component given to the user including all the information kept in the necessary tables. Users are not required to enter data to the search fields. If the search button is clicked while all the fields are empty, then the system calls all the educational elements from the relevant tables as can seen in the Figure 3.34.

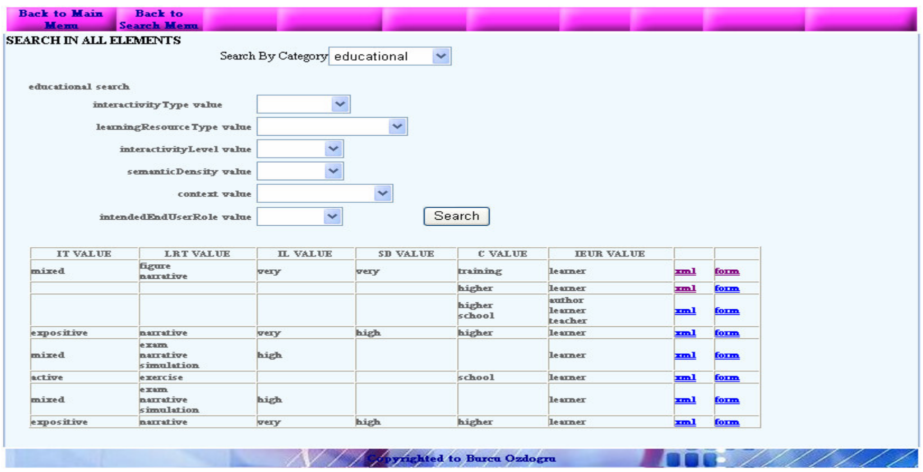


**Figure 3.34 – Search Results of Educational Category Search**

From the results table seen in the Figure 3.34, users can either choose xml or form view of the component they select and examine the views with the support of whole database.

If users are doing search by using the search fields of the nine category elements, they can simply enter the necessary data to the fields and by pressing "search" button they can reach the results. Also, in this option they are not required to enter data to the fields like the other search options and also they can view the component as xml or form. The xml notifications or form view give the whole learning object meta-data elements of the chosen component.

If the users fill the search fields, system searches the relevant tables and outputs the results in the table. According to their demand, users may do search either in all the fields or individual fields. As seen in the Figure 3.35, users ask the system to retrieve a query of a component which is related with humans, has a cost value, contributed by a publisher and described as a text. These fields are related with four different category elements so the mechanism searches the related tables joins them and if it finds a record shows it in the table. Users are not required to enter exact data, system does search based on the words entered to the fields. So, the "human" word entered into the title element, gives the results of all titles including "human".

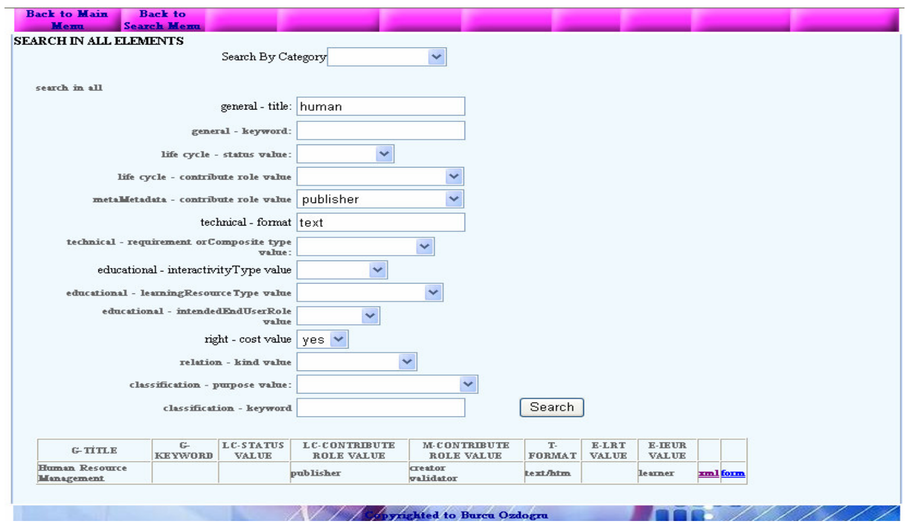


**Figure 3.35 – Search Result Based on Selected Search Fields**

The query returns only one result and users can examine the component by the help of xml view and form view. Figure 3.36 and Figure 3.37 shows the xml notification and form view of the result shown in the table.


```xml
<lom>
<general>
    <identifier>
        <catalog>POOL</catalog>
        <entry>000SamplePoolRefernece#</entry>
    </identifier>
    <identifier>
        <catalog>URI</catalog>
        <entry>http://www.algonquincollege.com/distance/certfs.html</entry>
    </identifier>
    <title>
        <string language ="en" >Human Resource Management</string>
    </title>
    <description>
        <string language ="en" >This course deals with personnel functions, including concepts, principles and practices;
techniques of personnel administration (roles, staffing, human resources</string>
    </description>
    <keyword>
    </keyword>
    <coverage>
    </coverage>
    <structure>
        <source> LOMv1.0</source>
        <value></value>
    </structure>
    <aggregationlevel>
        <source> LOMv1.0</source>
        <value></value>
    </aggregationlevel>
</general>

<lifeCycle>
    <version>
        <string language=" en">MGT2310</string>
    </version>
    <status>
        <source> LOMv1.0</source>
        <value></value>
    </status>
    <contribute>
        <role>
            <source> LOMv1.0</source>
            <value>publisher</value>
        </role>
        <entity> BEGIN:VCARD\nFN:Algonquin College END:VCARD</entity>
        <date>
            <datetime>2001-02-01</datetime>
            <description>
                <string language=""></string>
            </description>
        </date>
    </contribute>
</lifeCycle>
<metaMetadata>
    <identifier>
    </identifier>
    <contribute>
        <role>
            <source> LOMv1.0</source>
            <value>creator</value>
        </role>
        <role>
            <source> LOMv1.0</source>
            <value>validator</value>
        </role>
        <entity> BEGIN:VCARD\nFN:Fisher;Sue END:VCARD</entity>
        <entity> BEGIN:VCARD\nFN:POOL Project END:VCARD</entity>
        <date>
            <datetime>0000-00-00</datetime>
            <description>
                <string language=""></string>
            </description>
        </date>
        <date>
            <datetime>2001-02-01</datetime>
            <description>
                <string language=""></string>
            </description>
        </date>
    </contribute>
    <metadataSchema1>IEEE LOM 1.0</metadataSchema1>
    <metadataSchema2>Scorm v1.0</metadataSchema2>
    <language>en</language>
</metaMetadata>
<technical>
    <format>text/htm</format>
    <size></size>
    <location>http://www.algonquincollege.com/distance/certfs.html</location>
    <installationRemarks>
        <string language=""></string>
    </installationRemarks>
    <otherPlatformRequirements>
        <string language="en">Group Work is required for computer conferencing</string>
    </otherPlatformRequirements>
    <duration>
        <duration></duration>
        <description>
            <string language=""></string>
        </description>
    </duration>
</technical>
<educational>
    <interactivityType>
        <source> LOMv1.0</source>
        <value></value>
    </interactivityType>
    <interactivityLevel>
        <source> LOMv1.0</source>
        <value></value>
    </interactivityLevel>
    <semanticDensity>
        <source> LOMv1.0</source>
        <value></value>
    </semanticDensity>
    <intendedEndUserRole>
        <source> LOMv1.0</source>
        <value>learner</value>
    </intendedEndUserRole>
    <context>
        <source> LOMv1.0</source>
        <value>higher</value>
    </context>
    <typicalAgeRange>
        <string language="en">18--99</string>
    </typicalAgeRange>
    <difficulty>
        <source> LOMv1.0</source>
        <value></value>
    </difficulty>
    <typicalLearningType>
        <duration>
            <duration></duration>
            <description>
                <string language=""></string>
            </description>
        </duration>
    </typicalLearningType>
    <language>en</language>
</educational>
<rights>
    <cost>
        <source> LOMv1.0</source>
        <value>yes</value>
    </cost>
    <copyrightAndOtherRestrictions>
        <source> LOMv1.0</source>
        <value>yes</value>
    </copyrightAndOtherRestrictions>
    <description>
        <string language="en">Registration $332.85 Can. Contact Carole Smith, csmith@algonquinc.on.ca</string>
    </description>
</rights>
<classification>
    <purpose>
        <source> LOMv1.0</source>
        <value>discipline</value>
    </purpose>
    <taxonpath>
        <source>
            <string language=""></string>
        </source>
    </taxonpath>
    <description>
        <string language=""></string>
    </description>
</classification>
</lom>
```



**Figure 3.36 – XML View of the Selected Search Result**

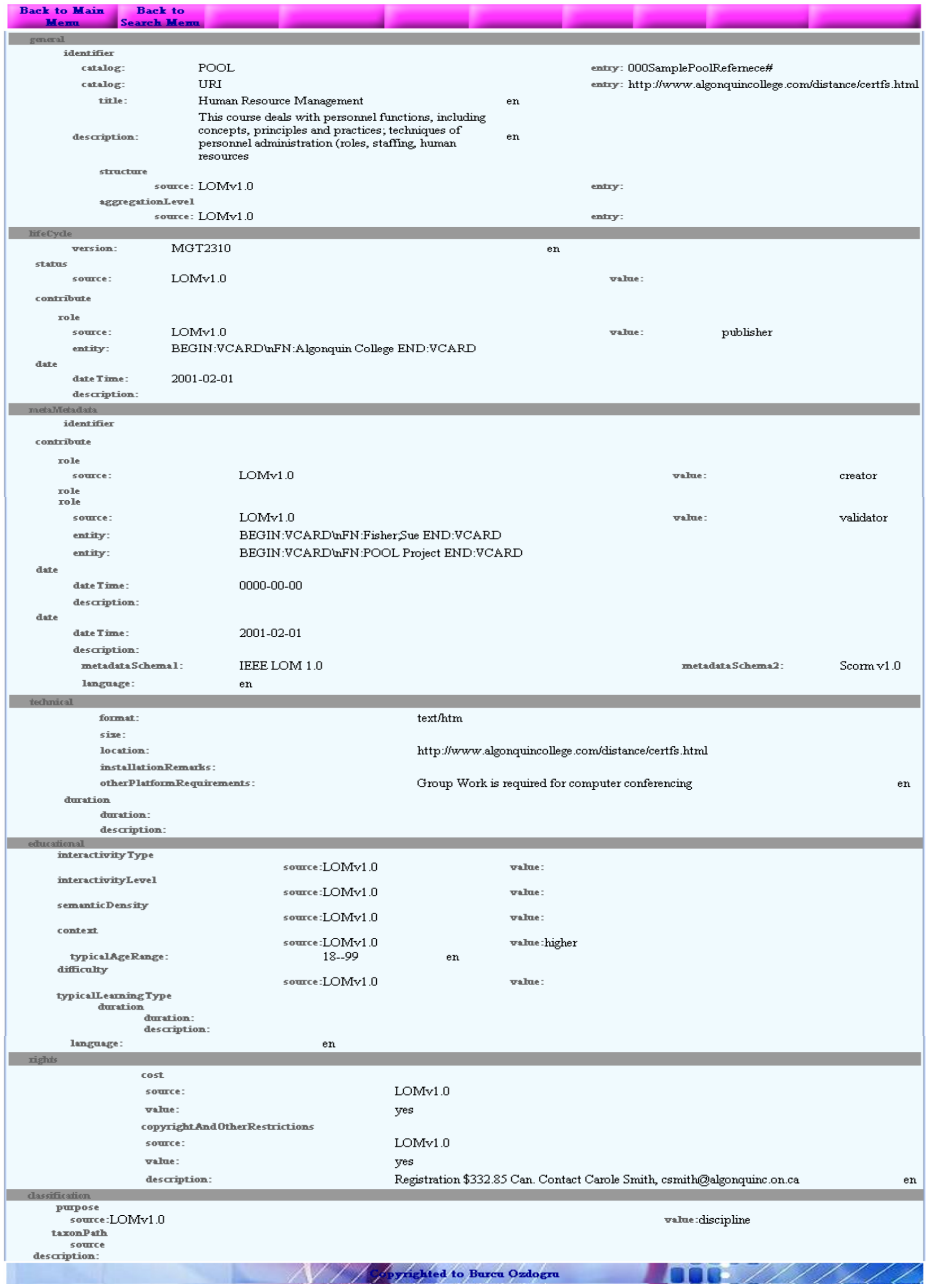


**Figure 3.37 – Form View of the Selected Search Result**

Through the complex queries done in this search module, the meta-data sharing between users becomes much easier. By the help of xml conversions users can use the output data for their own components exactly as it is or they can make changes by using an xml editor. Also, users can merge the data they got from the program with the data they have and build a new learning object meta-data by using metaXML.

# CHAPTER 4

# CONCLUSION AND FUTURE WORK

## 4.1 Conclusion

The aim of this study is to design a platform for educational meta-data. The metaXML provides a solution for the problems that arise from the structure of the meta-data and the deficiency of the newer technologies such as XQuery and XML databases. Since educational meta-data is a wide concept and since it accommodates different problems in its structure, metaXML gives a solution for the base of the whole system, thus when developing a whole system for educational learning the metaXML can be used as a starting point.

While developing such a system there were some problems faced during the design and implementation phases. First of all it is important to decide on the learning object meta-data standard to be used in the program. Since the standards Dublin Core, IMS and IEEE are similar in some ways it is hard to decide which one to use as a base in the program. While developing the system the method followed is to put the meta-data elements inside relational database for especially achieving complex queries.

According to the studies done in the area of XQuery and XML Databases it is possible to improve the system. It is a fact that keeping the meta-data in its own nature in a database may give a better performance. However, this kind of structure must be supported with XQuery for retrieving searches. Both XML Databases and XQuery are newer technologies compared to mature relational databases and structured query languages. After the improvements in this area, metaXML can be updated by using these technologies.

The structure of metaXML depends on SCORM Object Model however, XQuery depends on XML and XML Databases depend on the XML objects in the database. MetaXML provides relationships among objects but XQuery and XML Databases can not achieve this property. Another ability of metaXML is the capability of retrieving complex queries among multiple documents. This property cannot be achieved by using XQuery and XML Databases. Nevertheless, the semantics of metaXML is embedded in the database, but XML Databases cannot describe semantics, it just describes the structure and type names of the data.

As a conclusion, metaXML gives a different approach for developing an environment for educational meta-data. The only restriction of the metaXML is the stored semantic information in the database. If the structure of the XML standard will change in the future, this will affect the metaXML system as well. However, since the standards are mature enough the risk of this problem is lower in the educational domain.

## 4.2 Future Work

As mentioned before, this study aims to provide a solution for part of the problems that occur with educational meta-data. According to the package manifest this study interested in only the meta-data step but for developing a whole system, the other steps of the manifest must be developed by the programmers. Consequently, as a future work, the studies related with the educational learning will be continued by analyzing the other steps of the manifest. Nevertheless, it is possible to update the designed system due to the feedback expected to come after uploading the system into the Internet. MetaXML is developed as an open source document. We will also provide Turkish and English interface for metaXML. As a conclusion, since defining the meta-data of learning objects is the starting point for any educational organization, metaXML will help educators to create their own meta-data for the educational materials and start to share it among other educators.

# REFERENCES


[1] IMS Meta-data Best Practice Guide for IEEE 1484.12.1 – 2002 Standard for Learning Object Metadata, (2004).

[2] LEE T.B. Metadata Architecture, http://www.w3.org/DesignIssues/Metadata. (12.15.2005)

[3] Dublin Core Metadata Initiative, http://dublincore.org, (10.10.2005)

[4] IMS Global Learning Consortium, Inc., http://www.imsglobal.org, (10.10.2005)

[5] Draft Standard for Learning Object Metadata, (2002).

[6] IMS Learning Design Information Model, (2003).

[7] Walsh N. A Technical Introduction to XML, http://www.xml.com, (05.01.2006)

[8] W3C. Extensible Markup Language, http://www.w3.org/XML, (07.01.2006)

[9] Bourret R. XML and Databases, http://www.rpbournet.com/XML, (15.03.2006)

[10] Snelson J. All XML Databases are Equal, http://www.idealliance.org/proceedings/xtech05/papers/02-06-01, (02.01.2006)

[11] W3Schools. XQuery Tutorial, http://www.w3schools.com/xquery, (04.02.2006)

[12] World Wide Web Consortium, http://www.w3.org, (18.12.2005)

[13] Education Network Australia. The Edna Metadata Standard, http://www.edna.edu.au/edna/go/pid/385, (07.11.2005)

[14] Cover Pages. IMS Metadata Specification, http://xml.coverpages.org/ims.html, (05.01.2006)

[15] Learning Technologies Group. What is IMS, http://www.oucs.ox.ac.uk/ltg/projects, (21.10.2005)

[16] Advanced Distributed Learning, http://www.adlnet.org/scorm, (21.11.2005)

[17] Dr. Cathro W. Metadata: An Overview, http://www.nla.gov.au/nla/staffpaper/cathro3.html, (20.11.2005)

[18] Beavvoir P. Implementing IMS Metadata 1.1 Specifications as a Preliminary to Implementing Content Packaging 1.0 Specifications in Colloquia,

http://toomol.bangor.ac.uk/co3/documents/metadataimplementation.html, (24.12.2005)

[19] IMS Content Packaging Best Practice Guide, (2001)

[20] Dr. Wason T.D. Meta-Data Guide, http://www.twason.com/drtommeta.html, (26.10.2005)

[21] National States in Geographic Information Council. An Educational and Research Program in Support of Content Standards for Digital Geospatial Metadata, http://www.lic/wisc.edu/metadata, (10.10.2005)

[22] Kraan W. Educational Metadata, http://www.cetis.ac.uk/content2/20050210042132, (09.11.2005)

[23] SCORM 2004 Documentation, (2004).

[24] SCORM 2004 Conformance Requirements, (2004).

[25] Gatford M., Valsamidis T. Codd's Laws, http://www.soi.city.ac.uk/~tony/dbms/codds_laws.html, (28.04.2006)

[26] World Wide Web Consortium. XQuery 1.0 Formal Semantics, http://www.w3.org/TR/2001/WD_query_semantics_20010607, (09.04.2006)

[27] Wassilios KAZAKOS, Alexie VALIKOV, Andreas SCHMIDT, Rainer LEHFELDT, *Automation of Metadata Repository Development with XML Schema*, (2002).

[28] Roger S. PRESSMAN, *Software Engineering,* (2003)

[29] Greg RICCARDI, *Database Management with Web Site Development Applications,* (2003).

[30] Kanda RUNAPONGSA, *Methods for Efficient Storage and Indexing in XML Databases,* (2003).

[31] Shimin CHEN, Phillip B. GIBBONS, Suman NATH, *Stored Procedures for Distributed XML Databases,* (2004).

[32] Len SELIGMAN, Arnon ROSENTHAL, *The Impact of XML Databases and Data Sharing.*

[33] IEEE 1484.12.1/D1 Draft Standard for Extensible Markup Language (XML) Binding for Learning Object Metadata Data Model, (2003).

[34] Demetrios SAMPSON, Vicky PAPAIOANNOU, Panayiota KARADIMITRIOU, *EM$^2$ Environment for Editing and Management of Educational Metadata,* (2002).

[35] Eric DUVAL, Wayne HODGINS, Stuart SUTTON, Stuart L. WEIBEL, *Metadata Principles and Practicalities,* (2002).

[36] Zheng YIN, Zhengfang XU, Abdulmotaleb El SADDIK, *Study of Metadata for Advanced Multimedia Learning Objects,* (2003).

[37] Gottfried VOSSEN, Peter JAESCHKE, *Learning Objects as a Uniform Foundation for E-Learning Platforms,* (2003).